4th Dimension

Design Reference
Windows®/Mac™OS

4th Dimension Design Reference
For Mac™OS and Windows®

IMPORTANT LICENSE INFORMATION

Use of this software is subject to its license agreement included with the software. Please read the License Agreement carefully before using the software.

Contents

Preface 17

About the Manuals 18
About This Manual 19
Cross-Platform 19
Chapter Descriptions 19
Conventions 21

Chapter 1 4th Dimension Basics 23

Starting 4th Dimension 24
Creating a New Database 24
Opening an Existing Database 29
4th Dimension Desktop Files 32
Selecting a different data file 35
Linking a Data File to a Structure File 36
Making Backups 36
The Environments 37
The Design Environment 37
The User Environment 39
The Custom Menus Environment 40
Design Environment Editors 41
The Structure Editor 42
The Form Editor 43
The Method Editor 44
The Menu Bar Editor 45
The Password Access Editor 46
The List Editor 47
Picture Library 48
Compiler and application builder 50
Compiler 50
Application builder 51
Web Services Wizard 52

The Design Environment Interface 53
The Design Environment Menus 53
Contextual menus 60
Basics for Using Editor Windows 60
The Toolbar 62
The Explorer 63
Working with the Explorer 63
Tables Page 67
Forms Page 69
Methods Page 72
Constants Page 76
Commands Page 77
Lists Page 82
Components and Plug-ins Page 83
Using Comments. 84
Runtime Explorer. 90
Displaying the Window 90
Watch Page. 91
Process Page 93
Break and Catch Pages 93
Searching in the Database 94
Find window 94
String Types and Scope of the Search 96
Searching Options 97
Platform Interface 98
Where to set the platform 98
Platform Interface Settings. 99
4D Server Considerations 105

Chapter 2 Setting Preferences 107

Access to the Preferences. 107
Interface Theme 109
Look Page. 110
Formatting Page 113
Style sheet Page 114
Application Theme. 114
System Resources Page 115
Client/Server Page 118
Design Mode Theme 120
Fonts Page 120
Method Editor Page 121
Structure Editor Page. 125

Options Page 127
Documentation Page 129
Comments Page 131
Database Theme 132
Access Management Page 132
Data Control Page 134
Compilation Theme 137
Setting Page 137
Web Theme 141
Publishing Page 141
Configuration Page 145
4D WebSTAR Page 149
Web Services Page 150

Chapter 3 Designing a Database Structure 153

Database Basics 154
Tables 155
Creating a Database Structure 159
Using the Structure Editor 160
Selecting a Table Image 161
Scrolling the Field List 161
Resizing a Table Image 162
Moving a Table Image 163
Creating a New Table 164
Setting Table Properties 165
Creating Fields and Setting Field Properties 172
Creating New Fields 172
Field Types 176
Field Attributes 182
Choices and Help 188
Setting the Color of the Field 189
Modifying Fields and Field Properties 190
Renaming a Field 191
Changing a Field's Type 191
Changing a Field's Attributes 191
Indexing or Reindexing a Field 191
Relating Tables 193
Related Fields 195
The One Table and the Many Table 197
Setting Relation Properties 200
Related Fields 201
Many to One Properties 201
One to Many Properties 203
Wildcard Choice 204

Deletion Control 205
Color 206
Creating a Relation Between Tables 207
Configuring relations 208
Removing Relations 209
Reestablishing Relations 210
Modifying Relation Properties 210
Automatic and Manual Relations 210
Entering Data in Related Tables 211
Using Wildcard Choice Lists 212
Relation Types 214
One to One Relations 214
Many to Many Relations 215
Analyzing Database Relations 220
Circular Relations 221
Multiple Links to the Same Table 222
Relations from Multiple Records 223

Chapter 4 Creating Forms 225

About Forms 226
The Form Wizard 226
The Form Editor 227
Forms, Tables, and Fields 227
Active Objects and Graphic Objects 230
Graphic Objects 231
Object Properties 231
The Form Wizard 232
The Form Editor 234
Creating a New Form 235
Selecting Fields for the Form 237
Reordering Fields 240
Grouping Fields 241
Removing Fields 242
Using the Form Wizard's Advanced Options 243
Adding Fields 246
Customizing the Appearance of Form Objects 247
Customizing Buttons on the Form 250
Setting the Form Size 253
Field Label Placement 254
Form Display Options 254
Adding a Subform to the Form 255
Creating the New Form 257
Setting the Appearance of Form Objects 258
Fields and Field Labels 258

Using the Style Sheet Editor 259
Creating a Style Sheet 260
Applying a Style Sheet 262
Setting the Current Input and Output Forms 262
Deleting a Form 264
Renaming a Form 264

Chapter 5 Form Editor Basics 267

Using The Form Editor 268
Form Editor Window 268
The Tools Palette 273
Form Editor Menus 277
Displaying and Setting Form and Object Properties 282
Opening a Form in the Form Editor 288
Setting Form Properties 289
Naming the Form 289
Setting Form Access 291
Setting the Platform 292
Choosing a Form Type 293
Setting the Default Window Title 295
Assigning a Menu Bar to a Form 295
Settinizing and Resizing Options 297
Form Events 300
Contextual On-line Help 301
Managing Form Objects 302
Selecting Objects 302
Moving Objects 305
Resizing Objects 307
Using the Rulers 311
Creating Objects 312
Grouping Objects 315
Aligning Objects 316
Distributing Objects 322
Duplicating Objects 326
Copying Objects on a Form 328
Layering Objects 329
Deleting Objects 331
Optimizing the Appearance of Text and Picture Objects . . 332
Scaling a Form 333
Changing the Appearance of Objects 334
Platform Interface and Border Line Style 335
Working with Text Areas 336
Line Widths 341
Fill Patterns 342

Border Patterns . . . 343
Foreground and Background Colors . . . 344
Placing a Picture from the Picture Library . . . 345
Modifying the Background of the Picture . . . 346
Setting the Display Mode for a Static Picture . . . 347
Dissociating a form Picture from its Library Source . . . 349
Creating a Multi-page Form . . . 350
Adding a Display Page to a Form . . . 350
Moving from Page to Page . . . 352
Deleting a Page . . . 353
Adding Fields to a Blank Page . . . 354
Adding Page Navigation Controls . . . 354
Inherited Forms . . . 355
Using inherited forms . . . 355
Defining an inherited form . . . 356
Data Entry Order . . . 358
Viewing and Changing the Data Entry Order . . . 359
Setting the First Object in the Data Entry Order . . . 360
Using a Data Entry Group . . . 361
Restoring the Standard Data Entry Order . . . 361
Viewing and Printing Forms . . . 362
Saving Forms . . . 362

Chapter 6 Working with Fields and Active Objects . . . 365

Active Objects Defined . . . 366
Fields in a form . . . 366
Adding Fields to a Form . . . 367
Modifying a Field in a Form . . . 369
Changing a Field into a Variable and Vice-Versa . . . 370
Inserting Dynamic Table and Field Names . . . 370
Data Entry Controls . . . 371
Setting the Enterable and Mandatory Attributes . . . 373
Using Choice Lists . . . 374
Using Entry Filters . . . 376
Creating Custom Display Formats and Entry Filters . . . 385
Setting Maximum and Minimum Values . . . 388
Setting Default Values . . . 389
Keyboard layout . . . 392
Adding a Scroll Bar to a Text Object . . . 392
Adding Help to a Field or Object . . . 393
Display Formats . . . 396
Date Field Formats . . . 397
Time Field Formats . . . 398
Number Field Formats . . . 399

Alpha Field Formats 404
Boolean Field Formats 405
Picture Field Formats 409
Active Objects on a Form 412
Setting Object Properties 412
Creating an Active Object 414
Display Formats for Objects 415
Data Entry Controls for Enterable Objects 417
Setting the Tabable and Focusable Properties 417
Assigning a Keyboard Shortcut 418
Enabling Drag and Drop 419
Types of Active Objects 419
Enterable and Non-enterable Variables 420
Buttons 421
Check Boxes and 3D Check Boxes 426
Radio Buttons, 3D Radio Buttons, and Radio Pictures . . . 427
Picture Buttons 429
Pop-up Menus, Drop-down Lists, and Scrollable Areas . . . 435
Transparent scrolling areas 437
Combo boxes 438
Hierarchical Pop-up Menus and Hierarchical Lists 438
Button Grids 439
Picture Pop-up Menus 441
Tab Controls 445
Plug-in Objects 450
Indicators 453
Splitters 456
Duplicating on a Matrix 459
Using Object Methods with Fields and Objects 462
Object Events 463
Deleting an Object Method 467
Adding a Subform to the Form 468
Entering Data in a Subform 469
Data Entry Options for Subforms 470
Displaying Data in a Subform 471
Creating a Subform 471
Modifying a Subform source 476

Chapter 7 Output Displays and Reports 477

Output Forms for Listing Records 477
Output Control Lines 480
Moving Output Control Lines 483
Creating Output Forms 484
Using the Form Wizard's Advanced Options 486

Modifying an Output Form in the Form Editor 489
Displaying Several Lines Per Record 490
Modifying Output Forms for Reports 491
Printed Columnar Reports 491
One Record Per Page Reports 492
Using Subforms 492
Report with a Text Field 492
Custom Mailing Labels. 493
Creating Mail-Merge Documents 493
Basic Steps for Creating a Printed Report 496
Reports with Breaks 497
Creating Additional Control Lines. 499
Initiating Break Processing. 501
Reports with Subtotals 501
Summary Reports 505
An Example Report. 506
Printing Subforms, Pictures, and Text Fields 509
Printing Labels 512

Chapter 8 Creating Methods 517

4th Dimension Methods 518
Object Methods 518
Form Methods 519
Triggers 519
Database Methods 521
Project methods 522
Events. 522
Database Methods 524
Triggers 524
Form and Object Events 525
Introduction to Methods. 527
Examples 529
Where to Put an Object Method. 533
Creating or Opening Methods 534
Creating an Object Method 534
Creating or Opening a Project Method 535
Creating a Trigger 537
Creating or Opening a Form Method 538
Deleting methods 540
Deleting a Project Method, a Form Method or a Trigger . 540
Clearing Unwanted Object Methods 541
Defining the properties of project methods 541
Renaming a Project Method 542
Access and Owner privileges 543

Making a Method Invisible 544
Available through 4DACTION 544
Offered as a Web Service 545
Batch Setting for Method Attributes 545
Using the Method Editor 548
Using and configuring the Method editor window. 549
Writing a Method 559
Creating and using macros 577
Find and replace 582
Importing and exporting methods 585

Chapter 9 Creating Custom Menus 589

Designing Menus 589
Creating Menus 591
Basic Steps for Creating Menus 591
Default menu bars 592
Creating a Menu Bar 593
Adding Menus 595
Adding Menu Commands. 596
Rearranging Menus and Menu Commands. 598
Assigning Methods to Menu Commands 598
Working With Instances of a Menu 601
Creating Connected Menus 602
Modifying Connected Menus 602
Deleting Connected Menus 603
Enhancing Menus 603
Adding an Icon to a Menu Command 604
Changing Font Styles 604
Enabling and Disabling Menu Commands 605
Adding Separator Lines 606
Assigning Keyboard Shortcuts. 607
Deleting Menus and Menu Commands. 609
Previewing Menu bars and Adding a Splash Screen. 610
Menus and Custom Applications 611

Chapter 10 Managing Password Access. 613

Access System Overview 614
An Access Hierarchy Scheme 617
The Designer and the Administrator 619
Group Owners 620
Giving Users Design Environment Access. 621
Giving Access to the User Environment 621
Defining a Default User 622

Initiating the System 623
Assigning Users and Groups 624
Adding Users 624
Creating Access Groups 626
Assigning Users to Groups 627
Removing Users from Groups 628
Nesting Groups. 629
Removing Nested Groups 629
Saving and Loading Groups 630
Assigning a Group To Database Objects 631
Assigning Access To Record Operations 631
Assigning a Group to a Form 633
Assigning a Group to a Project Method 634
Assigning a Group to a Menu Command 635
Assigning a Group to a Plug-in or to the 4D Client Web server 636
System Maintenance 637
Administrator and Group Owner Access 638
Viewing Usage 638

Chapter 11 Creating Lists 641

Designing Lists for Data Entry 642
Hierarchical Lists 644
Required and Excluded Values 645
Non-Sequential Ranges of Values 645
Creating Lists 646
Adding Items to Lists. 647
Deleting Items and Lists 650
Adding a Small Icon to an Item 650
Adding a Reference ID to an Item 651
Specifying Ranges in a List 652
Sorting a List 652
Specifying Font Attributes 653
Making a Choice List Modifiable 653
Enabled/Editable option 655
Setting the Minimum Height of a List 656
Dragging a List into a Form 657

Chapter 12 Using the Picture Library 659

Managing and Viewing Pictures 659
Adding Pictures. 661
Importing a Picture File 661
Pasting a Picture from the Clipboard 662

Creating a New Picture 663
Setting Picture Properties 663
Creating and Modifying Pictures 664
Saving and Cancelling Modifications 666
Creating Thumbnails 666
Creating a Sequence of Frames 667
Size of the Frames 667
Viewing the Frames 668
Inserting and Deleting Frames 668
Shortcuts for Inserting Picture Buttons and Pop-up Menus 670

Chapter 13 Managing Processes 671

Processes 672
Processes Created and Managed by 4th Dimension 673
Time-Sliced Execution 674
Starting a New Process 675
Starting a New Process Using New Process 675
Starting a New Process Using the Menu Bar Editor 676
Starting a New Process Using Execute Method 678
Using The Process List 679
Process Number 681
Process Name 681
Process Status 683
Process Time 684
Controlling Process Execution 685
Pausing and Resuming a Process 685
Aborting a Process 685
Tracing a Process 686
Hiding a Process 686
Bringing a Process to the Front 687

Chapter 14 Compiling a Database 689

Introduction 689
What is a compiler? 689
Why compile your database? 691
Compilation in 4th Dimension 693
Compiler window 694
Compiler commands 695
Options 698
Compilation preferences 700
Compilation diagnostic aids 700
Symbol file 701
Error file 703

Range checking . . . 705
Diagnosing Anomalies . . . 707
Navigation between interpreted and compiled mode . . . 708

Chapter 15 Building Final Applications . . . 711

Application builder . . . 713
Definition of the application name and destination . . . 715
Building a compiled database . . . 716
Building a double-clickable application . . . 717
Selection of the 4D Engine folder . . . 717
Licensed version and demo version . . . 718
Generated files . . . 719
Rebuilding a double-clickable application . . . 719
Customizing the 4D Engine folder . . . 720
Customizing the application icon . . . 720

Chapter 16 Publishing or Using Web Services . . . 723

Presentation . . . 723
What are Web Services? . . . 723
Operation of Web Services —main definitions . . . 724
Integration of Web Services in 4D . . . 724
Publishing a Web Service with 4th Dimension . . . 728
Creating a Web Service method . . . 729
Publication of methods . . . 729
Generation of the WSDL . . . 731
Customizing a Web Service name . . . 732
Customizing a Namespace . . . 733
Adding comments to published methods . . . 734
Accessing a Web Service published by 4th Dimension . . . 734
Subscribing to a Web Service in 4th Dimension . . . 736
Principles . . . 736
Using the Web Services Wizard . . . 737
Display of connection parameters . . . 744
Calling a proxy method . . . 745
Processing complex types . . . 745

Appendix A Segmenting Data Files . . . 749

Segmenting Data Files . . . 749
Segmenting a Data File . . . 750
Limiting Data Segment Size . . . 751
Adding a Data Segment While Indexing . . . 752

Deleting Data Segments 753
Deleting a New Data Segment. 753
Deleting an Existing Data Segment 754
Missing Data Segments 754
Reconfiguring Data Segments 755

Appendix B Assigning a Help File to a 4D Database 757
Setting a Database On-line Help 758
File Formats 758
Assigning the Help File to the Database 758
Creating Contextual On-line Help 759
Calling the On-line Help from a Database 759

Appendix C DTD for macros. 761

Index. 763

Preface

4th Dimension is a powerful relational database application, a complete multi-platform development tool and a dynamic Web server.

You can use 4th Dimension to manage your own data or develop custom applications for different kinds of database management tasks.

For example, you can:

- Create a database structure of tables and fields,
- Design forms for entering, modifying, and displaying records,
- Search and sort records,
- Create reports and labels from information in the database,
- Import and export data between 4th Dimension databases and other applications,
- Publish your database on the World Wide Web.

With 4th Dimension, you can enhance conventional data management tasks with the following features:

- The powerful Form Wizard that lets you create sophisticated forms and reports with only point-and-click operations,
- A password access system to protect sensitive data,
- An integrated environment for generating charts that lets you produce a number of different chart types, extracted from the data,
- The capability to create custom applications with your own menus, dialog boxes, toolbars, and buttons; these applications can be compiled and distributed to other users,

- The possibility of using innumerable *Web Services* published on the Web within your database,
- A full-featured programming language that lets you incorporate commands or functions written in other languages.

Novice users can quickly create databases and begin managing their data. Experienced users can customize their databases with 4th Dimension's development tools. More experienced developers can use 4th Dimension's powerful programming language to add sophisticated features and capabilities to their databases, including file transfer, communications, and World Wide Web capabilities.

When you create a custom database, you can use custom menus and dialog boxes, button palettes, toolbars, and multiple windows to enhance your databases and make users more productive.

About the Manuals

The manuals described below provide a guide to the features of both 4th Dimension and 4D Server. The only exception is the *4D Server Reference* which describes features exclusive to 4D Server.

4th Dimension QuickStart leads you through the process of creating 4th Dimension databases. These examples provide hands-on experience to help you become familiar with the concepts and features of 4th Dimension and 4D Server.

The *User Reference* provides a description of the User environment — the built-in environment in which you can use databases to enter and manage data.

The *Design Reference* is a reference guide to the Design environment. It provides detailed descriptions of the operations that you use to build a database. You should use it in conjunction with the other volumes in your documentation package.

The *4th Dimension Language Reference* manual is the guide to 4th Dimension language. Use this manual to learn how to customize a database by incorporating 4th Dimension commands and functions.

The *4D Server Reference* is a guide to installing 4D Server and managing multi-user databases with 4D Server. This manual is included only in the 4D Server documentation package.

About This Manual

This manual describes the 4th Dimension Design environment. The Design environment is where you create a database before you enter data into records. This manual assumes that you have followed the instructions provided in the *Installation Guide* and that you are familiar with basic operations such as clicking, double-clicking, and choosing a menu command. To use the *Design Reference* effectively, you should do the following:

- Use *4th Dimension QuickStart* to work through the database examples as needed.
- Begin creating a database, referring to the *Design Reference* and *Language Reference* when you need to review a topic.
- Refer to the other manuals as needed.

Cross-Platform

This manual explains the use of 4th Dimension on both the MacOS and Windows platforms. Although the concepts and functionality of both versions of 4th Dimension are nearly identical, the manual addresses any differences where necessary. Such differences include the graphical user interface and keyboard commands.

This manual contains graphics illustrating the Windows environment. If there is a substantial difference in the appearance of the MacOS version of a window or dialog box, it is also presented.

Chapter Descriptions

This manual is divided into the following twelve chapters:

- **Chapter 1, "4th Dimension Basics,"** introduces basic 4th Dimension operations such as starting 4th Dimension, setting preferences, using the Design environment menus, and using the 4th Dimension Explorer. It also describes the 4th Dimension environments and provides an overview of the Design environment editors.
- **Chapter 2, "Setting Preferences,"** describes each modifiable parameter in the Preferences of 4th Dimension.
- **Chapter 3, "Designing a Database Structure,"** introduces the Structure editor and explains how to create tables, fields, and related tables.
- **Chapter 4, "Creating Forms,"** introduces the Form Wizard and explains how to create forms.

- **Chapter 5, "Form Editor Basics,"** explains how to use the Form editor to set form properties, create and modify form objects, and set object properties.
- **Chapter 6, "Working with Fields and Active Objects,"** explains how to set and enforce business rules for fields and enterable objects and how to add custom interface elements such as drop-down lists, hierarchical lists, tab controls, and picture buttons.
- **Chapter 7, "Output Displays and Reports,"** explains how to create a form for printing a report. It includes an explanation of how to create subtotals and other summary calculations using methods.
- **Chapter 8, "Creating Methods,"** introduces the 4th Dimension Method editor and explains how to use it to create methods.
- **Chapter 9, "Creating Custom Menus,"** explains how to use the Menu Bar editor to create custom menus. It also explains how to use connected menus to simplify menu management.
- **Chapter 10, "Managing Password Access,"** explains how to use the Password Access editor to create a system that controls access to tables, table operations, forms, methods, menu commands, and plug-ins.
- **Chapter 11, "Creating Lists,"** explains how to use the List editor to create lists.
- **Chapter 12, "Using the Picture Library,"** explains how to create and manage pictures for use as buttons, icons, and background images.
- **Chapter 13, "Managing Processes,"** introduces the concept of multitasking in 4th Dimension using processes. It explains how to start a process and how to use the Process List editor to view process information and control process execution.
- **Chapter 14, "Compiling a Database,"** details the integrated functions that can be used to compile your database.
- **Chapter 15, "Building Final Applications,"** explains the operation of the 4th Dimension application builder.
- **Chapter 16, "Publishing or Using Web Services,"** explains the advantages of Web Services and describes how to publish methods as Web Services or call external Web Services from within your databases.

- **Appendix A, "Segmenting Data Files,"** explains how to structure a data file so that you can store more than 2 GB of data or use more than one volume to store the data file.
- **Appendix B, "Assigning a Help File to a 4D Database,"** explains how to install a custom help file in a 4th Dimension database.
- **Appendix C, "DTD for macros,"** explains how to use the DTD (Document Type Declaration) for macros used by the Method editor.

Conventions

All the manuals in your documentation package, including this one, use certain conventions to help you understand the material.

The following explanatory notes are used:

Note Text emphasized like this provides annotations and shortcuts that will help you use 4th Dimension more productively.

4D Server Throughout the manual, 4th Dimension and 4D Server/4D Client are referred to simply as 4th Dimension. Differences between the operation of the two products are explained in 4D Server notes which provide information about using 4D Server/4D Client. This information is provided only when the operation of 4D Server/4D Client differs from that of 4th Dimension.

Warnings like this alert you to situations where data might be lost.

In addition, names of tables in a database are shown in brackets in the text to help distinguish them from the names of fields, forms, and other items. For instance, the Companies table is written as the [Companies] table.

Using Hypertext

If you are reading this manual on-line, you will find that the entries in the table of contents and the index are hypertext buttons. Hypertext links are printed in blue. Click a table of contents entry or a page number in the index to jump to that page. Throughout the body of the manual, cross-references to other sections are also hypertext buttons. Click a cross-reference to jump to that section and use Adobe Acrobat's **Back** button to retrace your steps.

1 4th Dimension Basics

This chapter provides basic information about 4th Dimension and the Design environment. The chapter includes the following:

- Instructions for starting 4th Dimension,
- Instructions for managing 4th Dimension desktop files,
- Instructions for backing up 4th Dimension databases,
- A description of the three 4th Dimension environments,
- An overview of the Design environment editors you use to create a database,
- An overview of the compiler and the stand-alone application generator,
- An overview of the Web Services Wizard,
- Instructions for navigation among the menus and windows of 4th Dimension,
- Instructions for using the 4th Dimension Explorer,
- Instructions for using the 4th Dimension Runtime Explorer,
- Instructions for using the Search editor of the Design environment,
- Instructions for choosing the platform interface,
- A description of the specific features related to 4D Server.

Unless otherwise noted, all instructions and explanations apply to both 4th Dimension and 4D Server.

Starting 4th Dimension

When you start 4th Dimension, you can either create a new database or open an existing one.

A 4th Dimension database consists of several desktop files. Under MacOS, the database is contained in two desktop files, the *Structure file* and the *Data file*. The Structure file contains all the specifications for the database and the Data file contains your data and any indexes you create. Under Windows, resource files for both the Structure and Data files are also created, making a total of four required files.

When you create a new database, you have an option to create a new folder that contains these files. When opening an existing database, you have an option to change the Data file that will be opened with the structure or create a new (blank) Data file that will be opened with the existing structure.

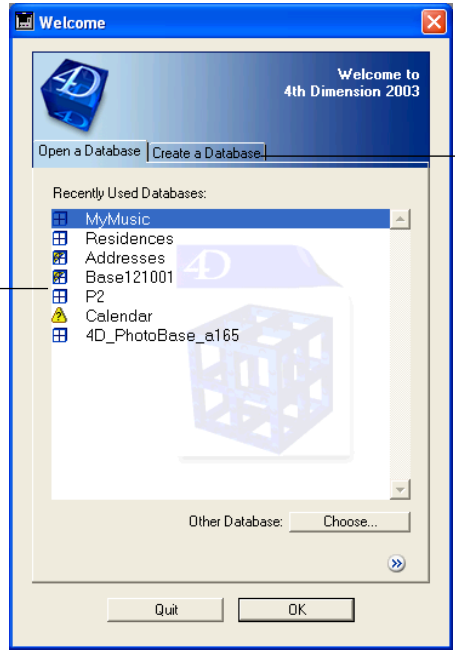
Note You cannot open 4th Dimension without opening a database and you cannot open more than one Data file or database at the same time.

Creating a New Database

If you want to start 4th Dimension and create a new database, follow these steps:

1 **Double-click the 4th Dimension or 4D Server application icon.**
OR
Select the 4th Dimension or 4D Server application icon and choose Open from the File menu.

The database opening and creation dialog box for 4th Dimension is then displayed:

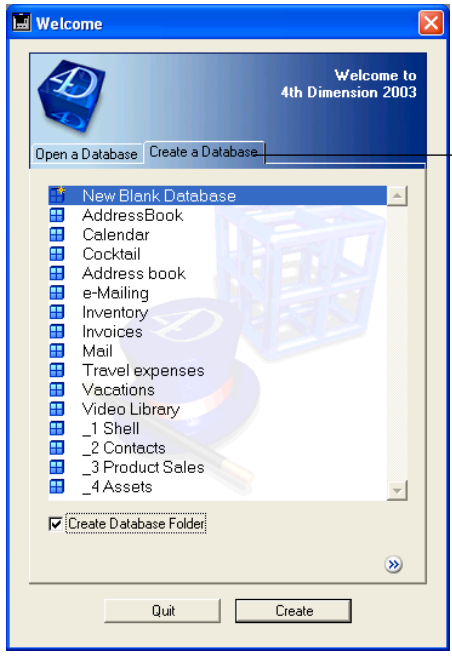


This dialog box lets you:

- Open an existing database. The “Recently Used Databases” list contains the names of databases that have already been opened on the machine. For more information, refer to the paragraph “Opening an Existing Database,” page 29.
- Create a new database. To do this, you must display the database creation page which can be accessed using the **Create a Database** tab.

2 Click the Create a Database tab.

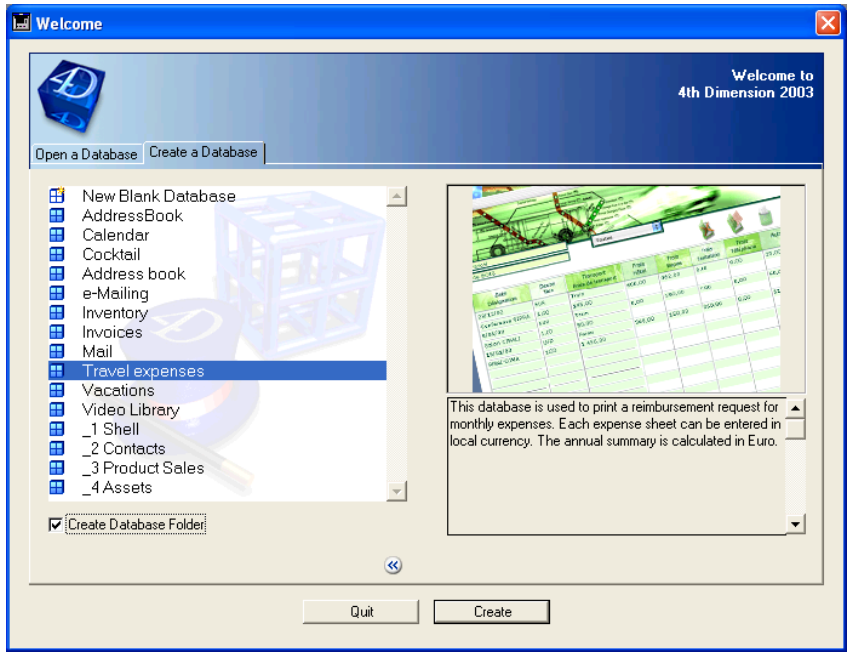
The following page appears:



You can either create a blank database, i.e., empty (option selected by default), or select one of the database templates provided.

3 **If you want to create a database using a template, click on one of the templates provided in the list.**

The right–hand part of the dialog box then displays a preview of the template's interface with a brief description of its functions:



The templates contain sample data and are directly operational. Each template comprises an interface, tables, fields, forms, methods, etc. that can serve as prototypes for customized development. The possibility of creating databases using these templates also enables novice users to visualize the capabilities of 4th Dimension better.

Note The templates provided will depend on your version of 4th Dimension. It is also possible to add to them (see the paragraph "Location of templates," page 29).

OR:
If you want to create a blank database, choose the "New Blank Database" option.

If you check the **Create Database Folder** option, all the database files will be created and placed in the same folder, which has the same name as the database.

4 **Click on the Create button.**

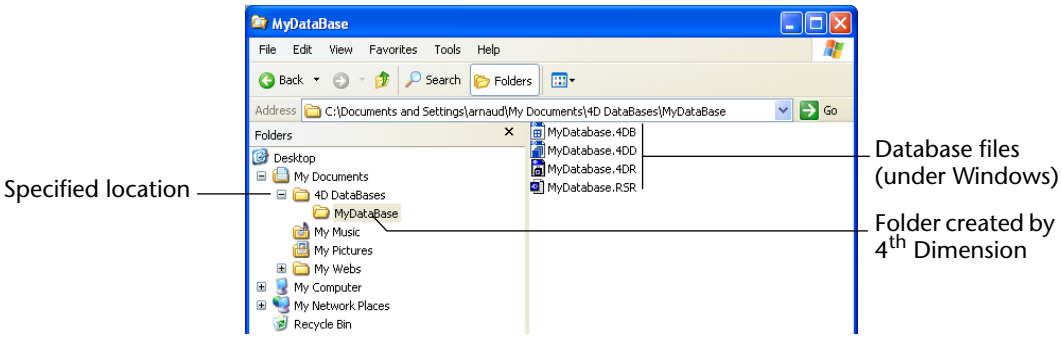
A standard Save file dialog box appears, allowing you to choose the name and location of the 4th Dimension database.

5 Type the name of your database and click on Save.

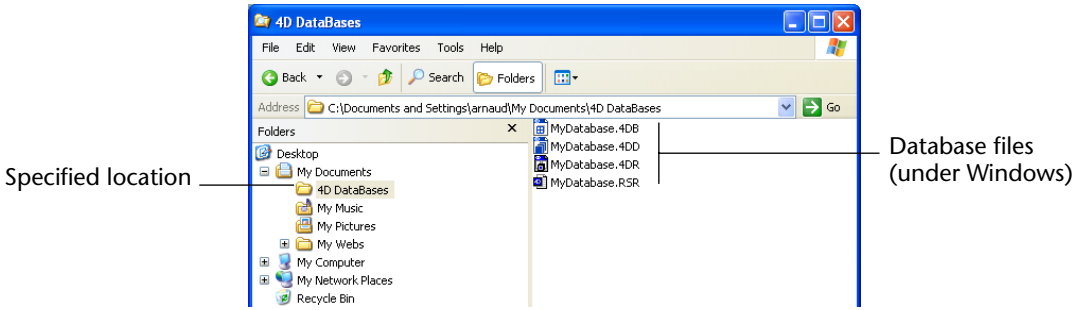
You can choose any file names that are accepted by your operating system.

When you validate the dialog box, 4th Dimension creates a new database having the name that you specified.

- If you checked the **Create Database Folder** option, the database files are grouped together in a folder having the same name as the database. This folder is located in the directory you specified in the Save-file dialog box:



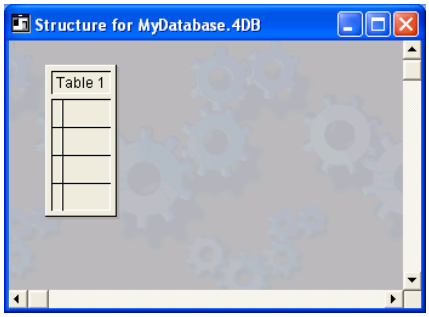
- If you did not select this option, the database files will be created at the location you selected.



4D Server The files are stored on the server machine when you first create the database using 4D Server. You then modify the database design by accessing the database from any client machine using 4D Client. When you open the database from the client machine, the Structure editor window shown in the figure below is displayed.

Once the database has been created, 4th Dimension displays either the Structure editor window of the empty database (if you created a blank database), or the customized splash screen of the database template used:

Blank database: empty Structure editor window



Template (example): splash screen of Travelling expenses template

You can then begin to use the database, change between environments, add fields and tables, etc.

Location of templates

Database templates are stored in a folder named “4D Templates” or “4D Modèles.” This folder must be placed next to the .exe file of the 4D application (under Windows) or next to the software package (under MacOS).
Additional templates will be provided from time to time by 4D. You simply need to add them to this folder in order for them to appear in the list of templates available.

Opening an Existing Database

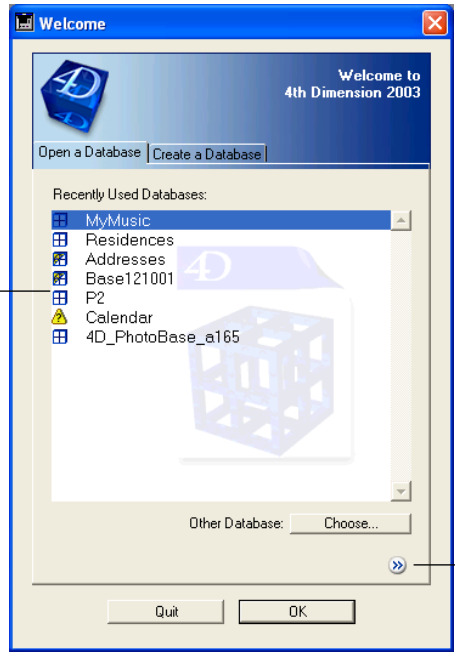
If you are starting 4th Dimension and want to open an existing database, follow these steps:

1 **Double-click the desired structure file (Under Windows, the .4DB file).**
OR
Drag the structure file icon on the 4th Dimension application icon.
The database opens.

OR

1 **Double-click the 4th Dimension application icon.**

4th Dimension displays the database opening and creation dialog box:



List of databases that have already been opened (empty when opening for the first time)

Button for displaying the opening options area

4th Dimension stores the names and paths of all the databases it opens. The databases are listed in chronological order, with the most recently opened database at the top of the list.
The list displays different icons according to the type of file referred to:

- Interpreted only structure file.
- Structure file comprising the interpreted and compiled code of the database[1]. The database is run in either the interpreted or compiled version according to the opening option chosen (see the paragraph “Opening options,” page 31).
- Compiled only structure file.
- Database with at least one file (structure, data or resources) missing (moved, renamed, deleted...).

2 To open a database that is displayed in the list, double-click its name
OR
Select its name and click the OK button.

1. For more information on the new principles concerning compilation in 4th Dimension version 2003, refer to Chapter 14, “Compiling a Database” on page 689.

The database opens.

Notes
- To delete the reference to a database from the list, select it and press the **Delete** or **Backspace** key.
- If you delete, move, or rename the database files using the file manager of your operating system, the reference to the database will be removed from the list.

3 **To open a database that is not available in the list, click Choose.**

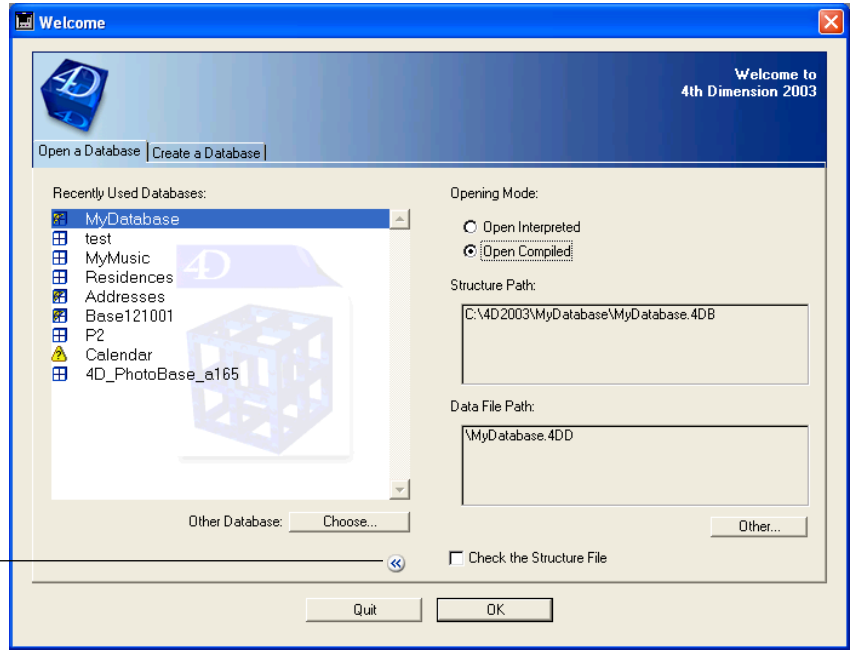
A standard open file dialog box appears, allowing you to select the structure file you want to open.

If a password is required, you are prompted to enter your password. The database opens in the environment specified in the Preferences dialog box. By default, the database opens in the Design environment.

4D Server For information about opening a database using 4D Client, refer to the *4D Server Reference* manual.

Opening options

The right-hand part of the database opening and creation dialog box (hidden by default) contains several opening options. To display the options area, click on the button :



- **Opening Mode**: This option can be used when a file containing both the interpreted and compiled code[1] of a database is selected in the left-hand area. You can then choose the opening mode via the **Open Interpreted/Open Compiled** radio buttons. If one of these two modes is not available (database not compiled or not recompiled after modification, compiled only structure file, etc.), the corresponding radio button is disabled.
  The option selected by default corresponds to the last mode in which the database was used.

Note 4D also allows you to switch from interpreted mode to compiled mode (and vice versa) during use. For more information, refer to the paragraph "Navigation between interpreted and compiled mode," page 708.

- **Structure Path and Data File Path**: This area indicates the names and paths of the structure and data files of the database. If the data file is comprised of several segments, this area will refer to the main segment (for more information about data segments, refer to appendix A, "Segmenting Data Files," page 749).
- **Other...**: Clicking on this button displays a standard Open File dialog box. You can then create or select a different data file for the database.
- **Check the Structure File**: When this option is checked, the logical integrity of the structure file (tables, forms, etc.) is checked when the database is next launched. In this case, a diagnostic window appears and indicates the state of the structure file. This functionality is identical to that offered by 4D Tools. For more information, refer to the 4D Tools *Reference Manual*.

4th Dimension Desktop Files

Under Windows, 4th Dimension creates four files for each database: a structure file, a resource file for the structure, a resource file for the data, and a data file. They are initially placed in the same directory when the database is created. You may place these files in different locations.

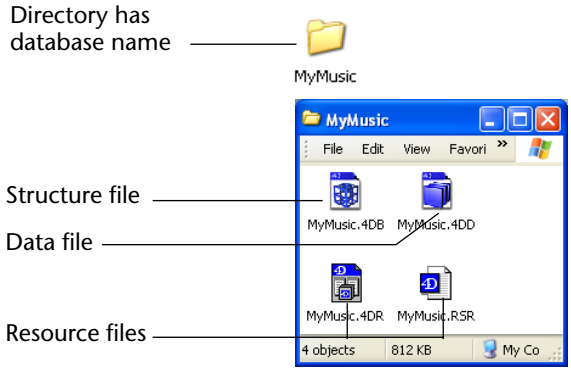
1. For more information concerning compilation in 4th Dimension, refer to the chapter "Compiling a Database," page 689.

Under MacOS, only the structure and data file are created; the resources are integrated in them.

The structure file contains all the specifications for the database structure (tables, fields, field properties), forms, methods, menu bars, password access groups, pictures, and lists. The structure resource file contains the MacOS resources associated with the structure file. The data file contains the data that has been entered into records and any record-dependent information such as indexes. The data resource file contains the MacOS resource files associated with the data file.

Under Windows, the DOS name of each file has an extension that identifies it as a structure, resource or a data file. The structure file has the same DOS name as the database, followed by ".4DB". The structure resource file has the same DOS name as the database, followed by ".RSR". The data resource file has the same DOS name as the database, followed by ".4DR".

The data file's DOS name is the database name followed by ".4DD".



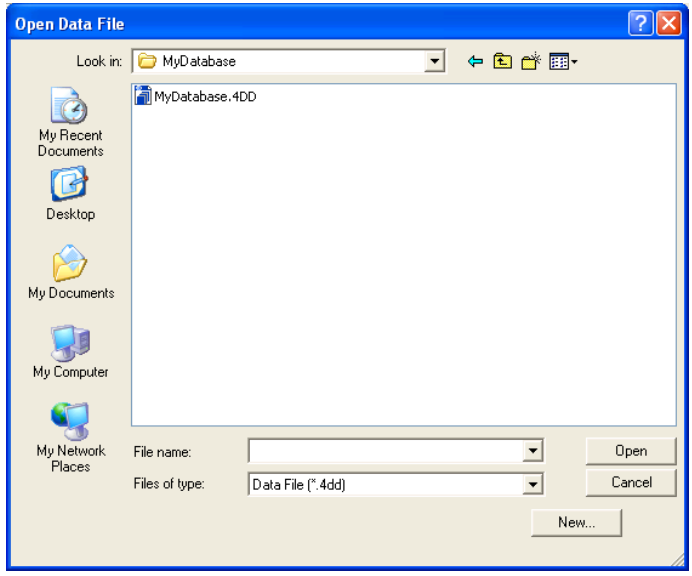
Note Under MacOS, the structure file does not have a suffix and contains both the design objects and the MacOS resources. There is no .RSR or .4DR file on the Macintosh. The data file has the suffix ".data" by default and contains both the data and their resources. The database folder, when it is created automatically by 4D, has the same name as the structure file, followed by an "*ƒ*".

In theory, you can use any data file with any structure file. The data file does not have to have the same name as the structure file, but the data contained in it must be compatible with the structure you want to use. That is, the data must fit into the fields; the number of fields in the structure must be at least as large as the number in the data file, and the data must agree with the field types.

Under Windows, the .4DB and .RSR files must have the same names and be in the same directory. Otherwise, you will not be able to open the database.

When you open a 4th Dimension structure file, the application opens the data file in the directory that has the same name as the structure file followed by ".4DD". If you change the name or location of the data file, the application will not be able to find the data file.

An Open data file... dialog box will appear to allow you to select the data file you want to use or to create a new data file.



Selecting a different data file

When opening a database, you can choose to select a different data file since it is possible to use different data files with the same structure.

To select a different or a new data file, you can either:

- press the **Alt** key (**Option** key under MacOS) when opening a database,
- or click the **Other...** button in the database opening options dialog box (see the paragraph "Opening options," page 31).

In both cases you can either create a new data file or open an existing one.

If you choose to create a new data file, 4th Dimension opens the database using the original structure but with no records.

When you use a different data file or create a new data file, 4th Dimension records the absolute path to that data file. The path specifies the location of a file. For example, if the data file "CONTACTS.4DD" is located in the CONTACTS directory in the WORK directory on the C volume, the path to the data file would be:

C:\WORK\CONTACTS\CONTACTS.4DD.

Once you have used the Open Data File dialog box to locate a data file for a particular database, 4th Dimension subsequently opens that same data file in the same path, unless it finds a data file with the same name as the structure file followed by .4DD located in the same directory as the structure file. If it does not find *StructureName*.4DD in the same directory as the structure file, it then attempts to locate the data file based on the path you gave it.

If you move or rename the data file again, you will need to locate it again.

Note Under MacOS, the same data file located on the hard disk named "Hard Disk" would be accessed by the path: Hard Disk:Work:Contacts:Contacts.data.

Linking a Data File to a Structure File

In most cases you would not want a user to be able to open a different data file. One important reason for this is that if the structure file is incompatible with the data file, the data file will be re-configured to match the structure file. To avoid this, use the WEDD resource to lock a data file to the appropriate structure file. Locking a data file with a structure file prevents the structure file from opening a data file with a different WEDD resource. It does not, however, prevent the structure file from opening a data file without a WEDD resource. For more information about the WEDD resource, refer to the 4D Customizer Plus documentation.

Making Backups

As you work on databases, it is essential to develop a consistent method of backing up your work. In rare cases, unexpected interruptions such as a power failure or computer failure can damage a database. Although 4th Dimension can often recover your database after such damage, having a backup is necessary insurance.

After working in the Design environment, you should make a copy of the structure file (the file with the suffix ".4DB") and the structure resource file (the file with the suffix ".RSR"). After entering or modifying data, you should make a copy of the data file (the file with the suffix ".4DD") as well as the data resource file (the file with the suffix ".4DR").

Note Under MacOS, the data file has the suffix ".data".

The data file changes as new records are added and old records are modified or deleted. If a database is used infrequently, with perhaps only a few changes each day, backing up the data file once a week or even less often is probably sufficient. If a database is used often, a more structured backup system is needed.

For example, you may want to use the following system:

1. **Make a backup at the end of each day (for example, on a shared volume). Keep at least the last five backups — one for each day.**
2. **At the end of the week, store the last backup permanently on a removable medium (such as a CD-Rom).**

A backup system like the one above ensures that you will be able to deal with most operating incidents.

You can also use the 4th Dimension plug-in, 4D Backup, to protect your database against damage. With 4D Backup, you create a backup of your database and a special log file that keeps track of changes to your database since the backup was made. If necessary, 4D Backup can restore the database to its state at a point in time before the damage occurred.

For information about creating a log file, refer to the *4th Dimension User Reference*. For information about using 4D Backup, refer to the documentation that comes with the plug-in.

The Environments

You work with 4th Dimension in three environments. Each environment is a system of editors, toolbars, windows, and pull-down menus that allow you to perform database operations. The three environments are:

- Design
- User
- Custom Menus.

The Design Environment

The Design environment is used to design and develop databases. All aspects of your database design are implemented in the Design environment. You use the Design environment to create tables and fields; define relations among tables; create forms for data entry, display, and printing; implement a password access system; create custom menus; attach methods to database objects; or set the general preferences of the application.

For example, you might want to keep track of information about each of the employees in a company. In the Design environment, you create an [Employees] table and add fields to that table to store employee data, such as the employee's name, job title, start date, and salary[1].

1. In the 4th Dimension documentation set, the names of tables appear in brackets. This is how table names appear in the Method editor.

You might also add a [Departments] table that contains information about each department in the company. You could then create a *relation* between these tables that allows users to view or modify department information from an [Employees] record and view or modify the list of [Employees] in a department from a [Departments] record.

You can use the Design environment to do the following:

- Create tables and fields in which to store data,
- Establish relations between tables,
- Create forms for entering, displaying, printing or publishing data,
- Create lists of choices that simplify and control data entry,
- Write methods to attach actions to database objects,
- Create custom menus and associate methods or standard actions to them,
- Create and manage multiple processes, allowing users to perform multiple database operations at the same time,
- Specify database preferences such as the default startup environment or the number of minutes between each automatic save of data,
- Set up a system of passwords to control access to information,
- Call or publish Web Services,
- Compile the database in order to accelerate its execution,
- Merge the database and the 4th Dimension engine in order to generate a stand-alone application.

Each of these features is described in this manual.

From the Design environment, you can switch to the User environment to try out your database structure or to the Custom Menus environment to view its customized operation. To do this, choose the desired environment from the **Use** menu.

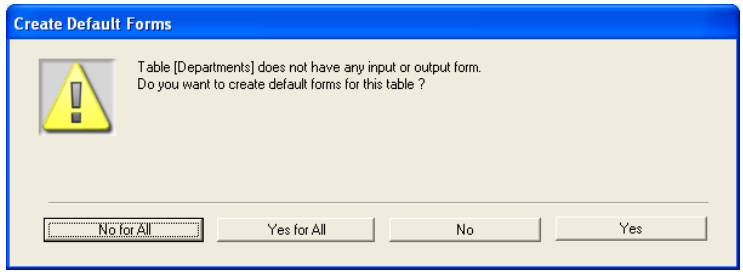
When you switch from the Design environment to another environment, the Design environment is still running and the its windows move to the background. If you want to hide the windows of the Design environment, hold down the **Shift** key while choosing the **User** or **Custom Menus** command.

The User Environment

The User environment is a generic end-user database application that you can use to enter and manage data. You can use this environment for tasks such as entering data, searching and sorting records, importing and exporting data, and printing reports and mailing labels. If you are in the process of developing a custom database application, you can use the User environment to enter or import a few records or test your methods before the custom database is finished.

The User environment includes generic editors that you need to import, enter, and export records, search and sort, and create reports, labels, and graphs. Also, the 4th Dimension programming language includes commands that let you incorporate the User environment's editors into your custom applications.

You can take advantage of the User environment to get a simple database up and running very quickly. To use the User environment, you only need to create your required tables and fields in the Design environment. If you switch to the User environment before you create forms for all the tables, 4th Dimension will ask you whether you would like it to create them automatically.



Since the User environment has its own editors for all standard database functions, you do not have to build any of these functions yourself.

The User environment allows you to do the following:

- Enter and modify data,
- View and print data,
- Search and sort records,
- Create reports and labels,
- Create graphs,
- Import and export data,

- Execute methods,
- Start processes,
- Publish a database on the World Wide Web,
- Work with any 4th Dimension plug-ins installed in the database.

From the User environment, you can switch to the Custom Menus environment or return to the Design environment by choosing either **Custom Menus** or **Design** from the **Use** menu.

The Custom Menus Environment

The Custom Menus environment is the environment that you use to run a custom application — an application that uses 4th Dimension but has its own menu system and screen design. The application behaves exactly as if it were deployed and running on 4D Server and 4D Client. Use the Custom Menus environment to preview the operation of a custom application that will eventually be deployed.

Note The custom menus environment can be used only if at least one menu was created using the Menu Bar editor. By default, 4th Dimension creates a standard menu bar.

In a custom application you control everything in the application, from the menus and forms it uses, to the methods used to accept, process, and display data. You are responsible for providing menu items and associated methods or standard actions that manage basic tasks such as data entry and modification, searching and sorting, and reporting. You can utilize any or all of the generic tools from the User environment or create your own screens and editors[1].

The Custom Menus environment can be completely different for each application you create. From the user's standpoint, the Custom Menus environment is a complete application for a specific kind of information management.

1. You cannot use the User Environment's Import and Export data dialog boxes in custom applications.

The default menu bar generated by 4th Dimension includes a **Use** menu allowing you to return to the User or Design environment.

Note If you have windows from more than one environment open at the same time, you can switch between environments by clicking their respective windows. When you click on a window, 4th Dimension places this window in the foreground and makes it the active environment.

Design Environment Editors

4th Dimension editors are used to create and modify the various components of your database design. Each editor is dedicated to one aspect of a design.

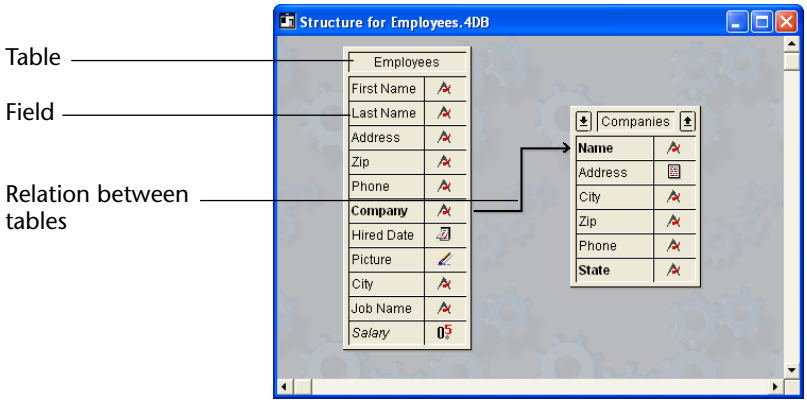
The Design environment contains the following editors:

- Structure editor
- Form editor
- Method editors
- Menu Bar editor
- Password Access editor
- Picture Library.

Each editor has its own window with appropriate tools, menu commands, and toolbars.

The Structure Editor

The Structure editor is your starting point for all design operations. Use the Structure editor to create tables, fields, and relations between tables. The Structure editor displays the images of the tables in the database and graphically shows the relations (if any) among the tables. The following illustration shows the Structure editor window.



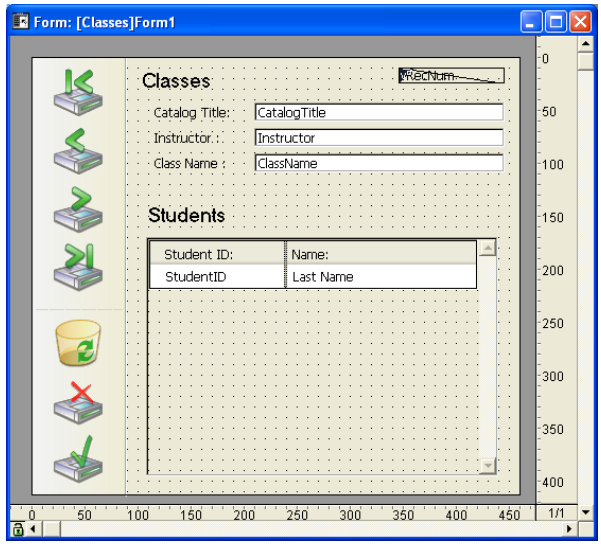
Use the Structure editor to:

- Create tables and subtables in a database,
- Specify table properties,
- Create fields and subfields,
- Specify field types and properties,
- Relate tables and set relation properties,
- Establish access privileges to tables and fields,
- View the structure of a database.

See Chapter 3 for more information about using the Structure editor.

The Form Editor

Use the Form Wizard to create forms and the Form editor to modify forms — both on-screen forms and printed reports. The Form editor lets you create sophisticated forms for managing your data.



Use the Form editor to:

- Add fields and variables to a form,
- Add buttons, combo boxes, drop-down lists, tab controls, and other interface objects to forms,
- Specify display formats and entry filters for data displayed and entered on the form,
- Add methods that enforce business rules during data entry or manage interface elements,
- Add graphic objects to the form — including text, lines, rectangles, and ovals — or paste in pictures or graphics that were digitized or created in another application,
- Specify fonts and font styles for objects containing text,
- Specify drag and drop actions,
- Specify automatic resizing and repositioning actions,
- Establish access privileges for forms.

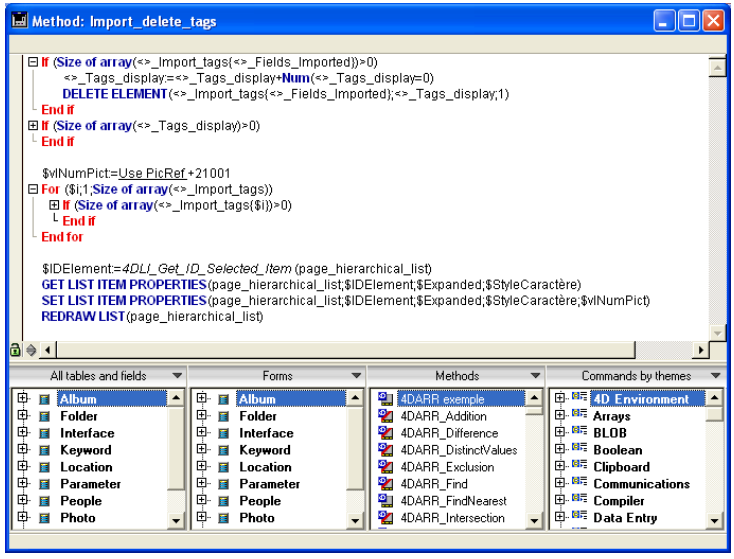
Refer to Chapter 4 and Chapter 5 for detailed information on how to create a form with the Form Wizard and how to modify forms with the Form editor. Refer to Chapter 6 for information about using the Form editor to control data entry and add interface elements to data entry forms. Refer to Chapter 7 for information about using the Form editor to design output forms, reports, mail-merge documents, and forms for mailing labels.

The Method Editor

Methods are instructions that process data or perform actions. Methods can perform tasks such as calculating the values of fields, transferring data from one table to another or validating data as it is entered. Methods also perform actions such as sorting, displaying, or printing records.

4th Dimension provides a Pascal-like language for writing methods via a Method editor and a library of commands and functions.

Here is a window of the Method editor.



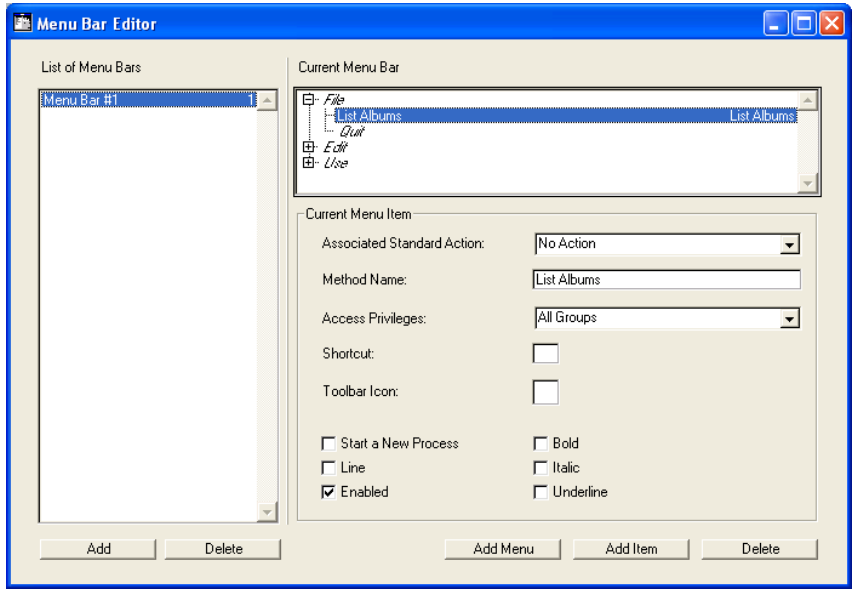
You can use the Method editor to:

- Write database methods that run automatically when particular worksession-related events occur,
- Write triggers that run automatically when particular database engine events occur,
- Create form methods that run automatically when a form is used,
- Create project methods that can be attached to custom menus, called by other methods, or executed by users in the User environment,
- Create object methods that are associated with fields or other form objects.

Refer to the *4th Dimension Language Reference* manual for detailed information about writing methods.

The Menu Bar Editor

When you create custom applications with 4th Dimension, you use the Menu Bar editor to create menu bars, menus, and menu commands. You can also attach menus to any form that you use for data entry. The figure below shows the Menu Bar editor being used to create a menu bar.



Use the Menu Bar Editor to:

- Create menu bars,
- Create and modify custom menus and menu commands,
- Attach icons to menu commands to create a custom toolbar,
- Assign project methods or standard actions to menu commands,
- Create connected menus that allow you to use the same menu definition in several menu bars,
- Preview the menus and menu bars as they will appear in the custom application,
- Include graphics for splash screens that display with menu bars,
- Establish access rights to menu commands,
- Set keyboard equivalents for menu commands,
- Enable or disable menu commands,
- Start a new process from a menu command.

Refer to Chapter 9 for a detailed discussion of adding custom menus and menu bars to your applications.

The Password Access Editor

4th Dimension lets you add passwords so that you can control access to aspects of your databases. The figure below shows the Password Access editor.

Use the Password Access editor to:

- Create users and give them passwords,
- Allow designated people to add users and change user passwords,

- Place users in groups,
- Give groups access to parts of the database, such as the Design environment and particular forms, menu commands, methods, and plug-ins,
- Monitor database use by individual users.

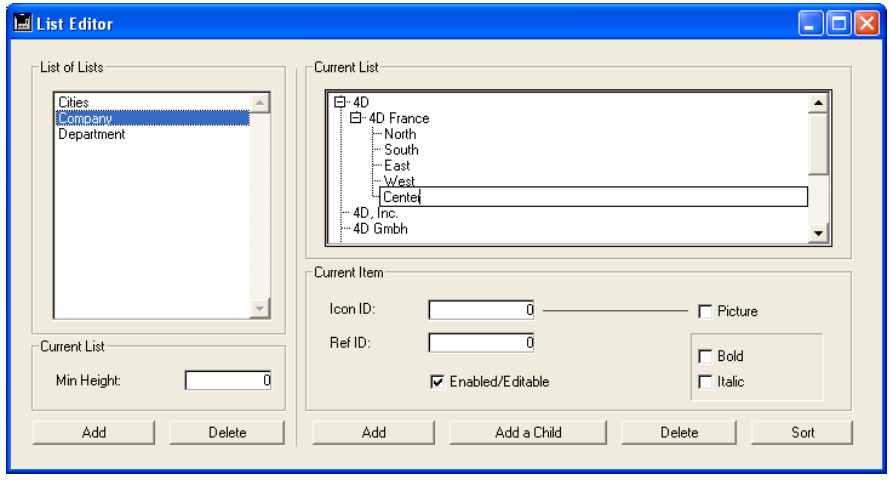
Refer to Chapter 10 for a detailed discussion of the Password Access editor.

The List Editor

You use the List editor to create lists. Lists can be used for several purposes in a database. Here are the most common uses:

- You can attach a list to a field. A user can select an entry from a choice list instead of typing it. With a choice list, you prevent entry of misspelled words and incorrect data.
- You can use a list to specify the items in pop-up menus, drop-down list boxes, combo boxes, scrollable areas, or tab controls.
- You can create hierarchical lists to populate hierarchical lists, hierarchical pop-up menus, or tab controls. These objects are then automatically defined using the list values.
- You can access the elements of a list in your methods or transfer the elements of an array to a list (and vice versa). By doing that, you can use a list to populate pop-up menus, drop-down list boxes, combo boxes, scrollable areas, or tab controls.

The figure below shows the List editor being used to create a list.



Use the List Editor to do the following:

- Create choice lists,
- Add items to a choice list,
- Attach small icons to list items,
- Delete choice lists or individual items in a list,
- Sort choices in a list,
- Activate values for use in tab controls,
- Make values editable for use in a hierarchical list,
- Make a choice list modifiable by the user.

Refer to Chapter 11 for a detailed discussion of lists.

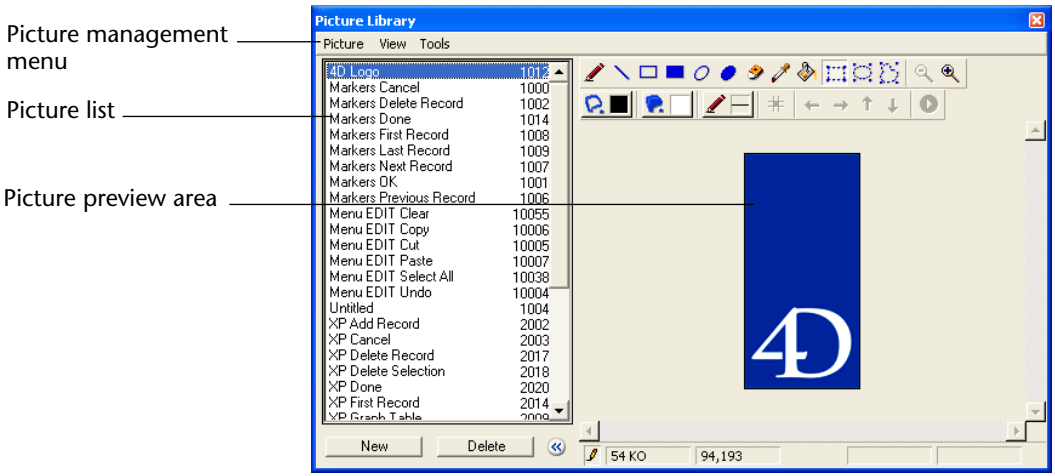
Picture Library

Use the Picture Library to store graphics that you can use as design elements in forms, as toolbar or list icons, picture menu items, or picture buttons. With the Picture Library, you can use a graphic in several places in your database but you need to store it in only one place. When you update an image in the Picture Library, all references to the image are updated automatically. This feature can reduce the size of your Structure files and make changes to the database easier to manage.

The Picture Library also includes a 2D Paint editor that allows you to create or retouch pictures. It is an ideal environment to design buttons and icons.

In addition to the features mentioned above, the Picture Library also includes integrated functions that can create or edit row by column arrays of pictures that are used to create picture buttons or picture menus.

The Picture Library is shown in the following illustration.



The Picture Library is used to:

- Store and preview the pictures of the database structure,
- Create or import new pictures,
- Retouch pictures,
- Define and preview tables of thumbnails,
- Drag and drop pictures to the Form editor, the List editor, or the Menu Bar editor,
- Delete pictures.

For more information about the Picture Library, refer to Chapter 12. For more information on inserting a picture in the Form editor, refer to "Placing a Picture from the Picture Library," page 345, "Picture Buttons," page 429, "Picture Pop-up Menus," page 441, "Adding an Icon to a Menu Command," page 604 and "Adding a Small Icon to an Item," page 650.

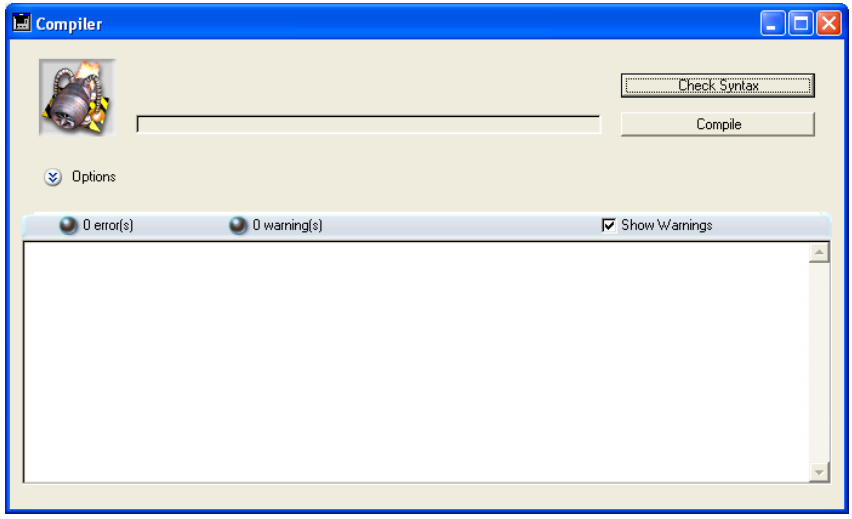
Compiler and application builder

Compiler

4th Dimension has an integrated compiler that can translate the entirety of the database code into machine language. Compiled databases run more quickly because 4th Dimension does not have to interpret the code each time a method is executed.

Note For more information about the principles of compilation, refer to the paragraph “What is a compiler?,” page 689.

Compilation is executed from the “Compilation” dialog box that can be accessed using the **Compiler** command of the **Tools** menu.



Note The **Compiler** menu command is disabled if the database does not contain at least one method.

This window is used for:

- Launching the compilation of the database (only with single-user 4th Dimension),
- Checking the syntax of the methods,
- Displaying or hiding the *warnings*,
- Building/rebuilding database typing methods,
- Deleting the compiled code.

Notes

- Database compilation requires an appropriate license. Without this license, it is not possible to carry out a compilation (the **Compile** button is disabled). Nevertheless, it is still possible to check the syntax and generate typing methods.
- With 4D Client, only the **Check syntax** and **Generate (typing methods)** buttons are active.

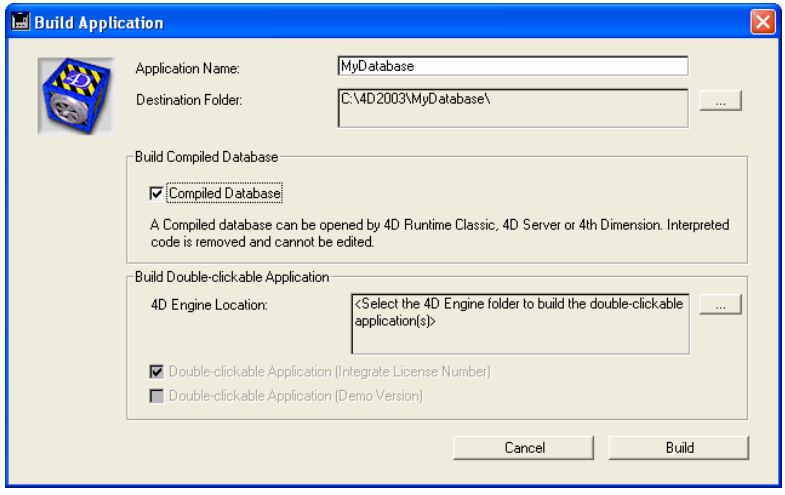
Once the database has been compiled, you can choose the running mode — compiled or interpreted — on startup of the database (see the paragraph "Opening an Existing Database," page 29) or via the **Use** menu (see the paragraph "The Use Menu," page 57).

For more information about the compiler, refer to the chapter "Compiling a Database," page 689.

Application builder

Once a database has been compiled, you can use the **application builder** to create either a compiled version of the database (without interpreted code), or a double-clickable version integrating 4D Engine, the 4th Dimension database engine.

To open the window used for building applications, choose the **Build Application...** command in the **File** menu. The following window appears:



Note If the database has not yet been compiled, an alert appears indicating that you must compile it.

The application builder allows you to:

- build a compiled database (without interpreted code) from your database. A database without interpreted code is protected since it is impossible to read or modify the code.
- build a double-clickable licensed application. A double-clickable application integrates the structure of your database and 4D Engine, the 4th Dimension database engine,
- build a double-clickable demo application.

For more information about the application builder, refer to the chapter "Building Final Applications," page 711.

Web Services Wizard

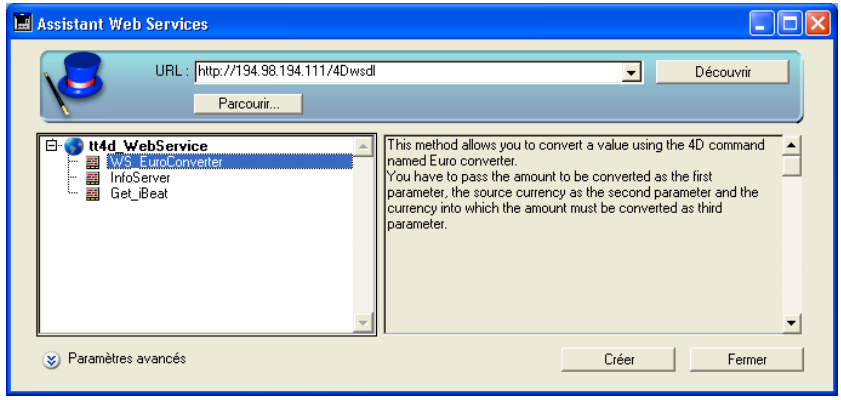
4th Dimension lets you publish or use *Web Services* within your databases.

A *Web Service* is a set of functions grouped together as an entity and published on a network such as the Internet. These functions can be called and used by any application compatible with Web Services and connected to a network. Web Services can carry out all sorts of tasks, such as supervising the routing of packages at a transporter's, e-commerce transactions, monitoring market values, etc.

The program publishing the services is called the "server." Any application compatible with Web Services can therefore use one or more of these functions; this is the "client" program. 4th Dimension can be used both as a Web Services client and/or server.

Note For more information about the *Web Services*, refer to the chapter "Publishing or Using Web Services," page 723.

The Web Services Wizard is used for the “client” part. It allows 4th Dimension applications to connect to servers offering Web Services and to generate the project methods needed for interrogating these servers:



The Web Services Wizard allows you to:

- connect to a server providing Web Services,
- parse and display the contents of the description file (WSDL),
- generate the “proxy method” needed for connecting to and interrogating the server of the Web Services.

For more information about the Web Services Wizard, refer to the paragraph “Subscribing to a Web Service in 4th Dimension,” page 736.

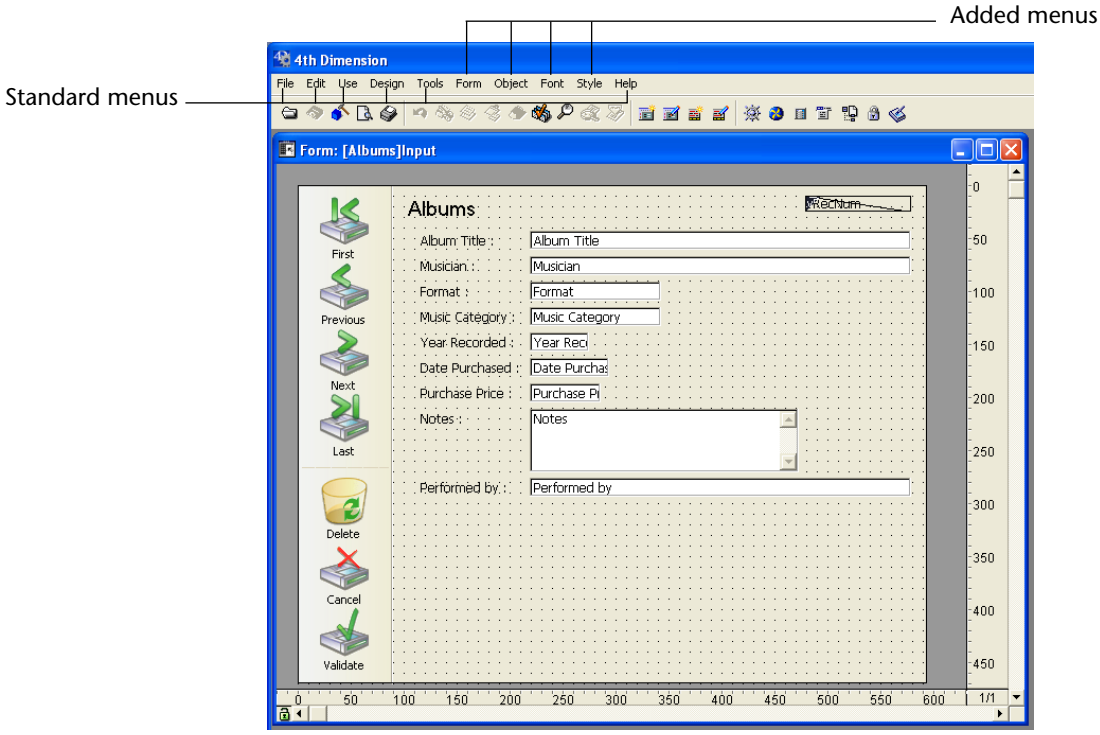
The Design Environment Interface

In the Design environment, you interact with 4th Dimension using its menus, contexual menus, toolbars, and windows. This section describes how to use these interface elements.

The Design Environment Menus

In the Design environment, the first five menus starting from the left are always the same, no matter which editor you are using — except for the picture library, which includes its own menu bar. Additional menus are added to the right of these standard menus for each of the editors. These menus provide additional commands for the editor that you are using.

The screen below shows the menus added when the Form editor is opened.



When several editor windows are open, the menus belonging to the frontmost editor window appear. You choose menu commands from these menus as you would in any application. For complete instructions for choosing from a menu, see the documentation that came with your computer.

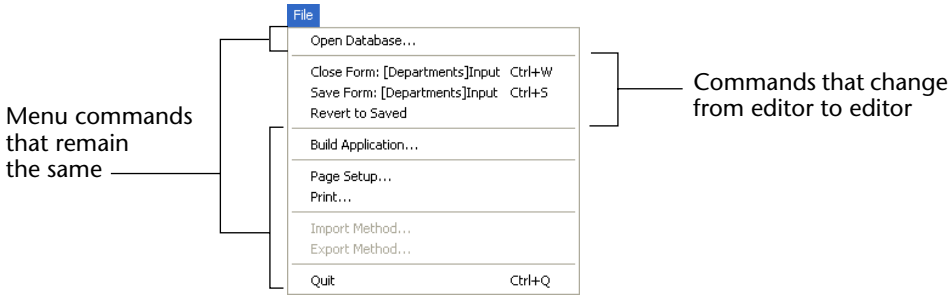
The Design environment has five permanent menus:

- File
- Edit
- Use
- Design
- Tools

The Help menu is managed by the operating system. Additional menus are active for different editors.

The File Menu

In the Design environment, the **File** menu provides standard file operations.



These File menu commands remain the same for every editor:

- **Open Database** Displays the database opening and creation dialog box. You can open an existing database at any time. 4th Dimension automatically saves changes to the current database before opening the next database.
- **Build Application...** Allows you to access the dialog box used for building the final application. This dialog box can only be accessed if the database has been compiled. For more information, refer to the chapter “Building Final Applications,” page 711.
- **Page Setup** You can set the specifications for printing at any time. Page setup information is stored with each form.
- **Print** You can print the contents of any editor window at any time. Printing will thus depend on the active editor.
- **Import method...** Allows you to import a method stored in the form of a text file. This command is only active when a window of the Method editor is in the foreground.
- **Export method...** Used to export the current method in the form of a text file. This command is only active when a non-empty window of the Method editor is in the foreground.
- **Quit** You can quit 4th Dimension at any time. 4th Dimension automatically saves your work before quitting.

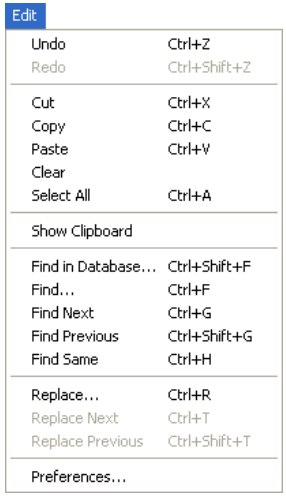
Note for MacOS X Under MacOS X, the **Quit** menu command is placed in the Application system menu.

These **File** menu commands change depending on the active editor:

- **Close *EditorName*** You can close editor windows at any time. 4th Dimension saves the contents of each window before closing it. If the Structure editor window is frontmost, choosing **Close Structure** in the **File** menu closes the Structure editor window. To close all the windows of the Design environment, hold down the **Shift** key while selecting the **User** or **Custom Menus** in the **Use** menu.
- **Save *EditorName*** You can save the contents of an editor's window at any time without exiting 4th Dimension. 4th Dimension automatically saves the contents of an editor when you close its window, change to a new environment, or exit the application.
- **Revert to Saved** You can revert to the last saved version of a form or method. This menu command replaces the contents of the Forms or Method editor with the last version saved.

The Edit Menu

In the Design environment, the **Edit** menu of 4th Dimension provides standard editing operations, find/replace commands as well as access to the application Preferences.



These are the menu commands provided by the **Edit** menu:

- **Undo** Use **Undo** to cancel the last operation carried out. This menu command is useful when you make a mistake and want to redo something.

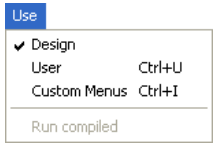
Note In the Method editor, 4th Dimension allows you to undo the last 20 actions carried out successively.

- **Redo** Use **Redo** to repeat the last operation that was cancelled. This command is only used in the Method editor.
- **Cut, Copy, Paste** You can select something on the screen and either cut or copy it. In either case, a copy of the selected object is placed on the Clipboard. You can then paste the object into a new location in the same window or in another window.
- **Clear** Use **Clear** to erase a selected object. No copy is placed on the Clipboard.
- **Select All** Use **Select All** to select every object in the current editor. For example, use **Select All** before adjusting all the elements of a form.
- **Show Clipboard** You can view the current contents of the Clipboard at any time. Sometimes you will want to view the Clipboard prior to pasting its contents.
- **Find in Database...** Use this command at any time to search for a character string, an object name, etc. among all the database objects (menus, forms, methods...). A window displays the search results.
- Find commands (**Find**, **Find Next**, **Find Previous**, **Find Same**) These commands are used to carry out searches among the methods. They can only be used in the Method editor.
- Replace commands (**Replace**, **Replace Next**, **Replace Previous**) These commands are used to carry out search-and-replace operations among the methods. They can only be used in the Method editor.
- **Preferences** You can configure the general Preferences of the 4th Dimension application at any time. For more information, refer to the chapter “Setting Preferences,” page 107.

Note for MacOS X Under MacOS X, the **Preferences** menu command is placed in the Application system menu.

The Use Menu

The **Use** menu lets you switch environments.

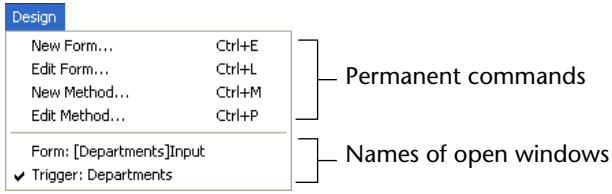


A check mark appears to the left of the current environment. To switch to another environment, choose it from the **Use** menu.

The **Run compiled** command is only active when the database has been compiled. It is used to run the database in compiled mode. By default, when the database is running in compiled mode, the command is renamed **Run interpreted**. For more information about the compiled mode, refer to the chapter "Compiling a Database," page 689.

The Design Menu

The Design menu has a divider that separates the permanent menu commands from the list of open windows:



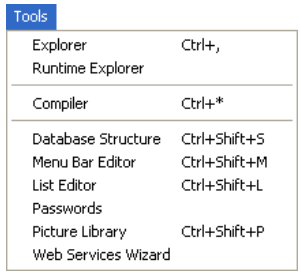
The four permanent commands let you create or edit forms and methods:

- **New Form** Opens the Form Wizard, ready for you to create a new form. For information about creating new forms with the Form Wizard, see "Creating a New Form" on page 235.
- **Edit Form** Opens the Forms page of the Explorer, ready for you select a form to open. For information about opening forms, see the section "Forms Page" on page 69.
- **New Method** Allows you to create a new project method. It first opens the "New method" dialog box so that you can enter the name of the new method. For more information about creating methods, refer to the paragraph "Using the Method Editor," page 548.
- **Edit Method** Opens the Methods page of the Explorer, ready for you to open an existing method. For information on opening methods, see the section "Methods Page" on page 72.

  The menu commands below the divider bring a Design environment window to the front. This list depends on the number and names of the open Design environment windows. A check mark indicates the active window.

The Tools menu

The **Tools** menu provides access to the Explorer, Runtime Explorer, Compiler, various Design environment editors, as well as the Web Services Wizard of 4th Dimension:



The **Explorer** menu command displays the Explorer. For information about the Explorer, see the section "The Explorer" on page 63.

The **Runtime Explorer** command displays the Runtime Explorer window. For more information about the Runtime Explorer window refer to "Runtime Explorer" on page 90.

The **Compiler** command displays the integrated compiler window of 4th Dimension. For more information, refer to the paragraph "Compiler window," page 694.

The following set of menu commands opens the corresponding editor windows of the Design environment. For information on each editor, see the following chapters or sections:

Editor	Reference
Structure	Chapter 3
Menu Bar Editor	Chapter 9
List Editor	Chapter 11
Password editor	Chapter 10
Picture Library	Chapter 12

Finally, the **Web Services Wizard** command displays the wizard window used to facilitate the integration of Web Services within your 4th Dimension database. For more information, refer to the paragraph "Subscribing to a Web Service in 4th Dimension," page 736.

Contextual menus

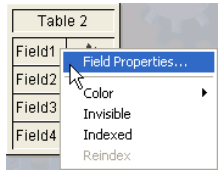
In most Design environment editors, you can use contextual menus to execute specific actions on objects or to open dialog boxes.

► To use a contextual menu:

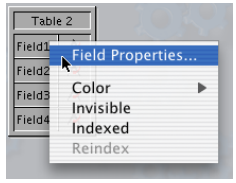
1 **Under Windows, click on an object or area with the right mouse button.**
Under MacOS, Control-click on an object or area.
The contextual menu associated to the object or area appears.

Contextual menu associated with the fields in the Structure editor

Windows



MacOS



Note Under MacOS, when you press the **Control** key, the pointer changes to a contextual menu pointer , indicating that a contextual menu is available.

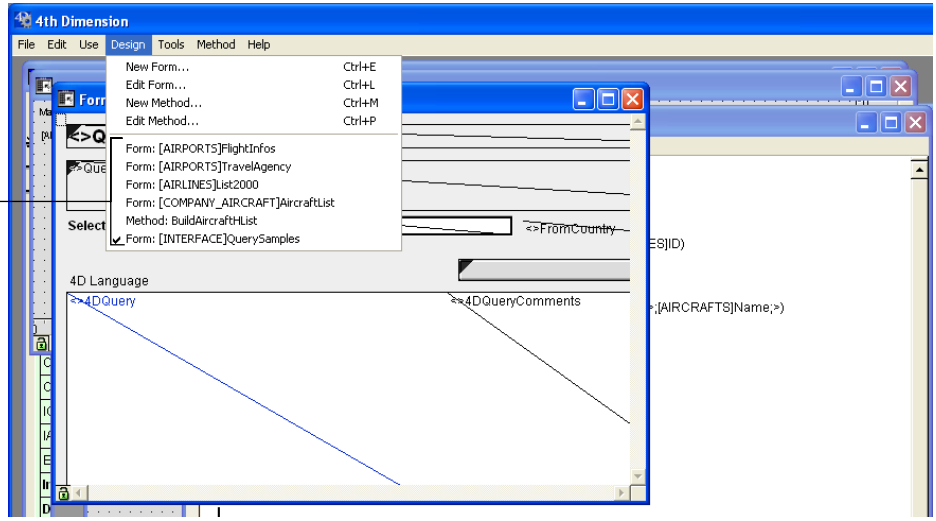
2 **Select a command in the contextual menu, as you would for any other menu.**

Commands in contextual menus vary according to the current editor as well as the object on which you clicked. As its name indicates, a contextual menu only includes menu items appropriate for a specific context. The contextual menu items perform functions that are identical to those in the standard menus, buttons, and keyboard shortcuts.

Basics for Using Editor Windows

Each 4th Dimension editor is displayed in a separate window. There may be several editor windows open at once. You can have more than one editor open at one time and some of the editors can display several windows at the same time.

A list of open windows is displayed at the bottom of the **Design** menu.



You can move between the open windows as you work. Only one editor window is active at any time. The active editor determines which menus are added to the right of the menu bar.

To make a window active, you can click anywhere in the window or you can choose the window's name from the **Design** menu. You can move a window anywhere on the screen by dragging its title bar.

You can expand the window to full-screen size by clicking the maximize box in the upper-right corner of the window. You can make the window any size you want by dragging the Size box in the lower-right corner of the window. Under Windows, you can drag the edge of a window to resize it.

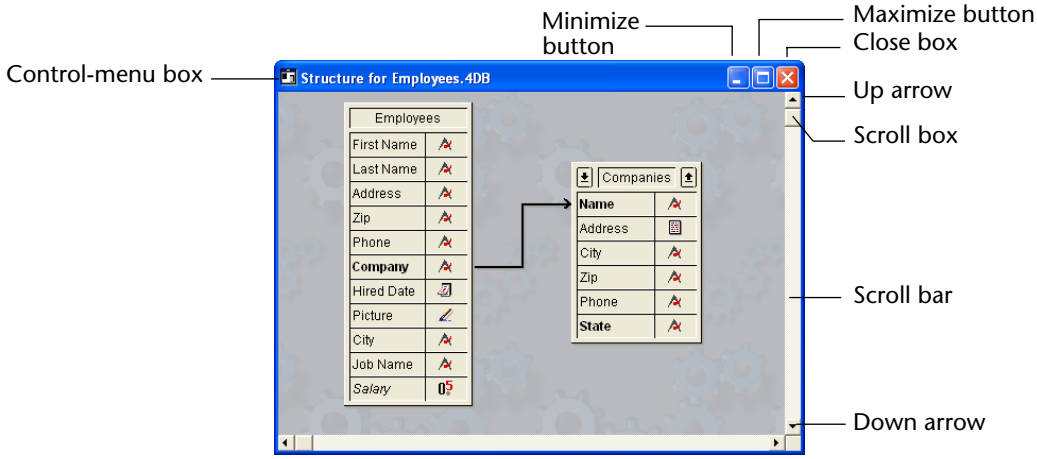
You close a window by clicking the Close box (MacOS) or double-clicking the Control-menu box in the upper left corner of the window (Windows), or by choosing **Close *EditorName*** from the **File** menu.

To close all the open windows of the Design environment, choose **Exit Design Environment** from the **File** menu.

To close all the Design environment windows except the Structure editor, press the **Alt** key while double-clicking the Control-menu box (Windows) or clicking the Close box (MacOS) of the frontmost editor window.

Scrolling

As you build a database, the contents of some of the editors can become so large that it is difficult to view the entire structure or form.



You scroll the contents of the window with the scroll bars as you would in any application.

The Toolbar

The Design environment displays a toolbar just below the standard menubar. The toolbar provides convenient access to all the Design environment menu commands.

The Toolbar contains buttons for all the standard menu items. The buttons are arranged in groups, with each group corresponding to a menu. Each Design environment editor has its own toolbar; the toolbar buttons correspond to the editor's menu commands. Toolbar buttons are grouped by menu and appear in the same left-to-right order as the menus.

When you are working with an editor window that adds commands to the standard Design environment menu bar, the corresponding buttons appear in the toolbar to the right of the standard toolbar buttons. The following illustration shows the Password editor toolbar of the Design environment:

To obtain help for a button, position the pointer over a button. The following illustration shows help for a Design menu button.

If you don't want to use the toolbar, you can hide it by deselecting the "Show Toolbar" option on the **Interface>Look** page of the application Preferences. For more information, refer to the paragraph "Look Page," page 110.

When you create custom menus, you can create custom toolbars and add buttons corresponding to your custom menu items. For information on creating custom toolbars, see the paragraph "Adding an Icon to a Menu Command," page 604.

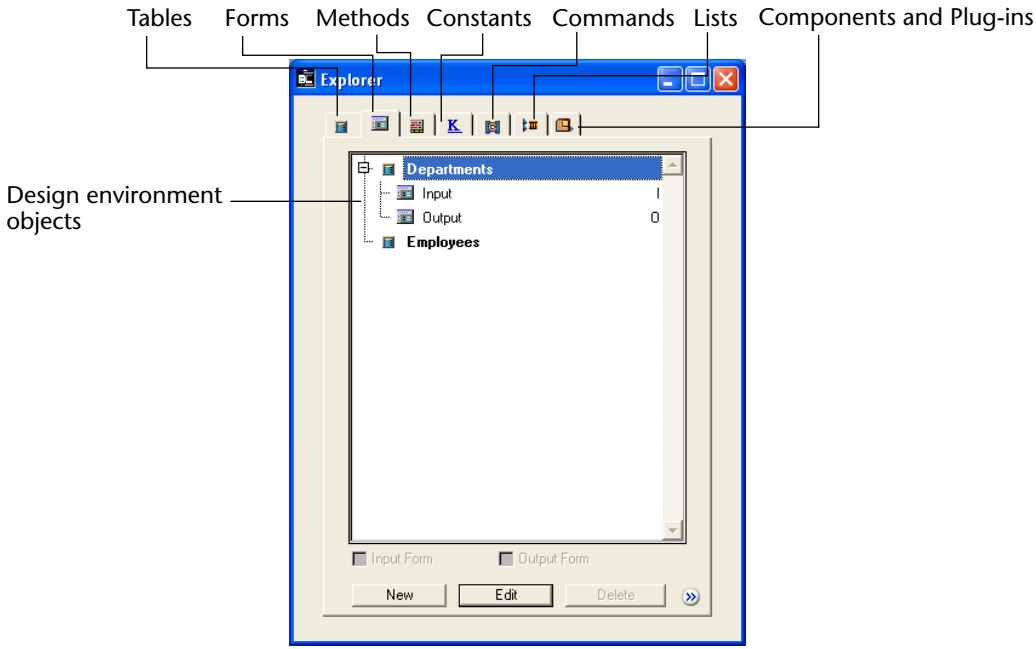
The Explorer

The Explorer is a window that gives you convenient access to tables, forms, methods, constants, built-in 4D commands, lists and components/plug-ins. You can display the Explorer at any time by:

- Choosing **Explorer** from the **Tools** menu,
- Pressing **Ctrl+Space bar** (Windows) or **Command–Space bar** (MacOS).

Working with the Explorer

The tabs at the top of the Explorer let you access different groups of Design environment objects. The Explorer has separate pages for tables, forms, methods, constants, commands, lists and components/plug-ins.



Displaying Different Explorer Pages

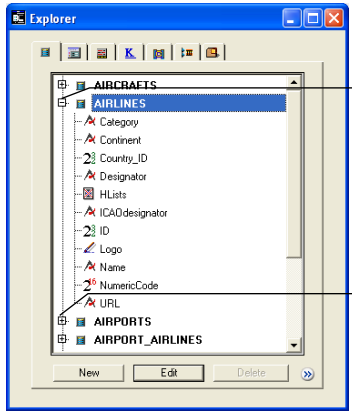
Click a tab to display either the Tables, Forms, Methods, Constants, Commands, Lists or Components and Plug-ins page. When you display a particular page, the appropriate Design environment objects are listed in the Explorer window. On all of these pages, the objects are displayed as a hierarchical list.

Expanding and Collapsing Hierarchical Lists

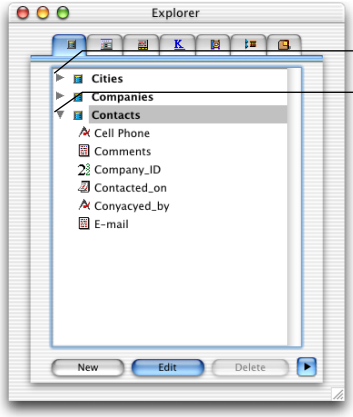
You can expand an object in the list by clicking the plus sign (Windows) or the triangle (MacOS) to its left or by highlighting the object and pressing the right arrow key. You collapse an expanded item clicking its downward-pointing triangle (MacOS) or minus sign (Windows) or by highlighting it and pressing the left arrow key. You can also expand or collapse an object by double-clicking the object.

The following illustrations show both expanded and collapsed topics:

Windows



Macintosh



Renaming a Form or Method

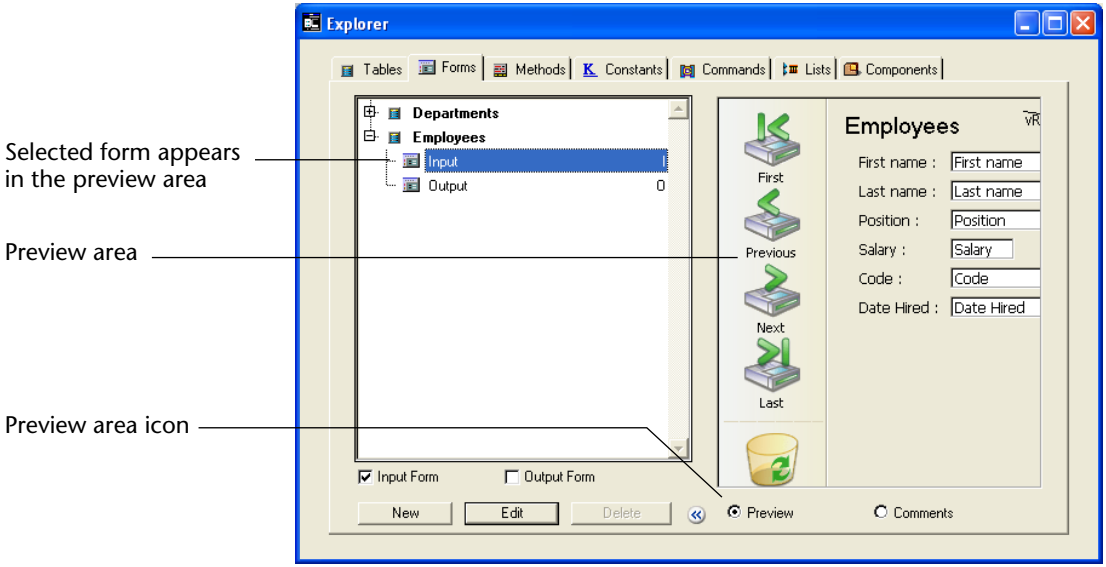
If you need to rename a form or method, hold down the **Command** key (MacOS) or the **Ctrl** key (Windows) and click the name of the form or method. The text becomes editable. Make your changes and then click anywhere outside the text area to save your changes.

The Explorer always lists forms and methods alphabetically. If the new name changes the sort order, 4th Dimension will resort the list when you click outside the entry area.

Displaying and Hiding the preview area

You can display or hide the Explorer preview area by clicking on the expand/collapse icon for the preview area . The preview window lets you preview table images, forms, methods, constant values, command documentation and lists, as well as information about the components and plug-ins installed in the database. You also can use the preview area to enter and view comments about certain database objects.

The following illustration shows a form being previewed:



To hide the preview area, click the preview area icon again.

Note When the preview area is shown, labels are added to the tabs. The labels automatically disappear when you hide the preview area.

Resizing the Explorer Window

You can resize the Explorer by dragging the lower-right corner of the window.

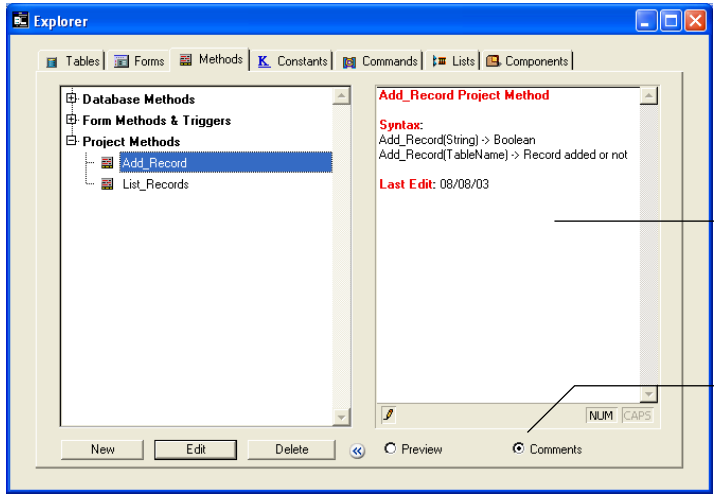
Note If the preview area is not displayed, you cannot display it by resizing the window. You must use the » icon.

If the preview area is not displayed, you can only expand the Explorer window vertically. When the preview area is displayed, you can expand the Explorer window both horizontally and vertically.

Displaying Comments

4th Dimension allows you to assign comments to the following database objects: methods (form methods, database methods, project methods, and triggers), forms, tables, and fields. Displaying and modifying comments are carried out using the Explorer (the preview area must be expanded).

To display and/or modify an object's comments, select the object in the object list and click the **Comments** radio button located below the preview area. The preview area is then replaced by the comments area.



The creation and use of comments are described in detail in the section "Using Comments" on page 84.

Using the Delete, New, and Edit buttons

The **Delete**, **New**, and **Edit** buttons below the list can be used to create, modify, or remove Design environment objects. These buttons are disabled automatically whenever an action is not possible. For more information on using these buttons, refer to the section on the appropriate Explorer page.

Using Drag and Drop

In many instances, you can use drag and drop to add a database object to an editor window. For example, you can add a field to a form by dragging a field name from the Tables page of the Explorer to an open form in the Form editor. When you are working with the Method editor, you can add the names of tables, forms, fields, project methods, constants, and commands as well as their syntax to a method using drag and drop.

The sections that describe each Explorer page give specific information on the drag and drop options for that page.

Each Explorer page is described briefly in the following sections. More detailed information is presented in the chapter that deals with the appropriate topic.

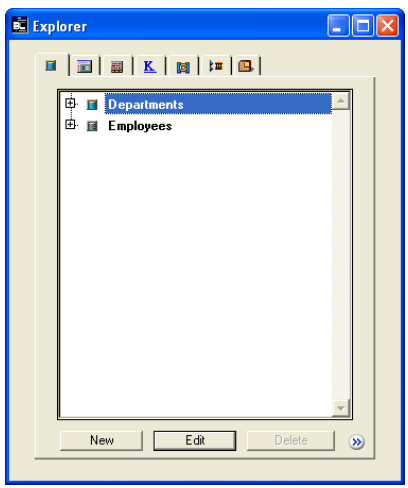
Using contextual menus

You can use contextual menus on the Forms and Methods pages of the Explorer. These menus provide access to additional functions. They are described in the following pages.

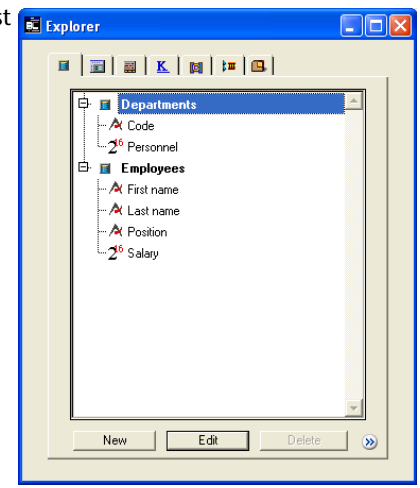
Tables Page

The Tables page lists all the tables and fields in the database. It can be used as an alternative to the Structure editor to access table and field properties. When a table is expanded, the fields in the table are shown.

Collapsed list



Expanded list



The field type is indicated by an icon or a letter to the left of its name. Double-click a field name to open the Field Properties window. For more information on field properties, see “Creating Fields and Setting Field Properties” on page 172.

Using Drag and Drop

You can add a field to a form by dragging the field name from the Tables page onto the form. You can add either a table name or a field name to a method by dragging the name to the method. When you do so, the name appears using the correct syntax. For example, if you drag the field “First Name” in a [Customers] table, it appears in the Method editor as “[Customers]First Name”.

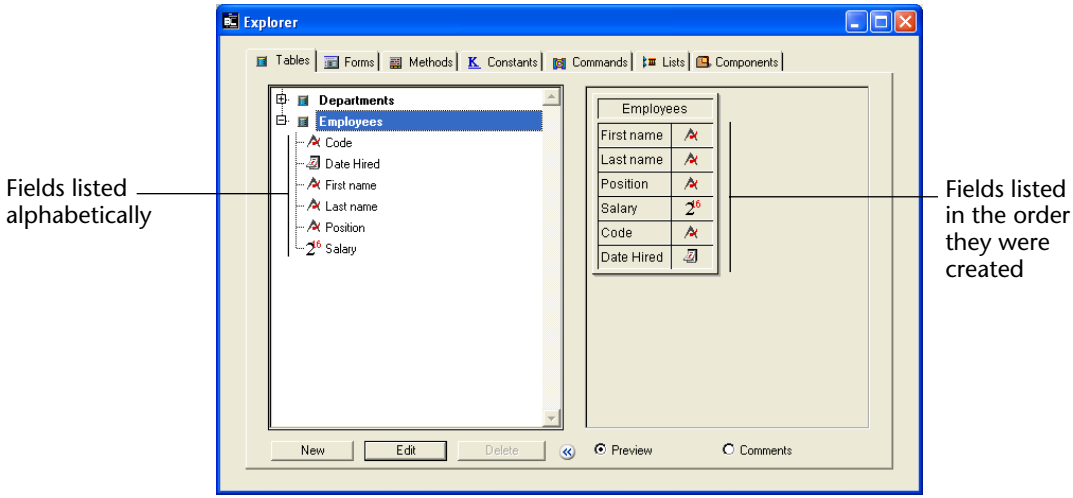
You can create a relation between two tables by dragging a field from the Tables page of the Explorer to the field that uniquely identifies the other table in the relation. For more information, see the section “Creating a Relation Between Tables” on page 207.

Viewing a Table Image

You can bring a a table image in the Structure editor window into view by double-clicking the table name in the Explorer. When you double-click a table name, 4th Dimension centers the table image you clicked in the Structure editor window. This feature is useful if you have a large structure and would otherwise need to scroll the Structure editor window to view a particular table image.

Previewing a Table Image

You can also preview the table image for a selected table. To preview a table image, highlight a table and click the Preview icon (if the preview area is not already displayed). The following illustration shows a table image being previewed.



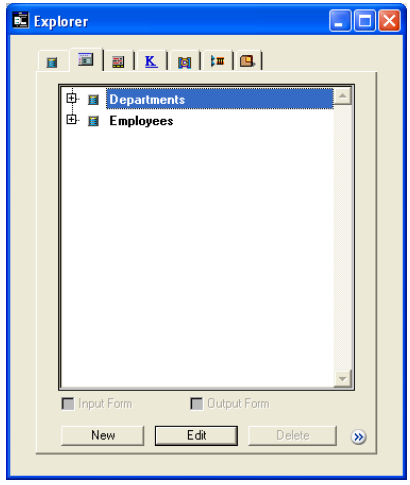
Setting Table and Field Properties

You can view or modify the properties of a table or field from the Tables page of the Explorer. Select a table or field, then click on the **Edit** button or simply double-click the table/field name. The Table or Field Properties window appears and displays the properties of the object concerned. For information on setting table and field properties, refer to the paragraph "Setting Table Properties," page 165 and to the paragraph "Creating New Fields," page 172.

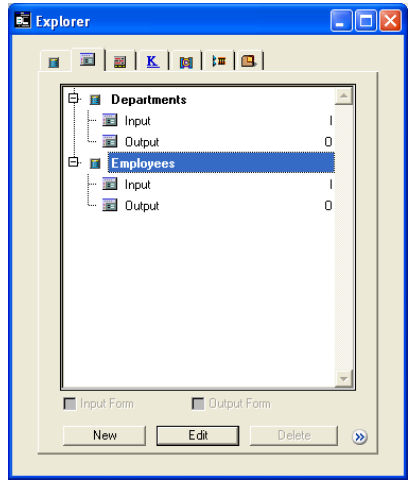
Forms Page

The Forms page lists all the tables and forms in the database. When a table is expanded, the forms for that table are shown:

Collapsed list



Expanded list



The Forms page is displayed automatically when you choose **Edit Form** from the **Design** menu.

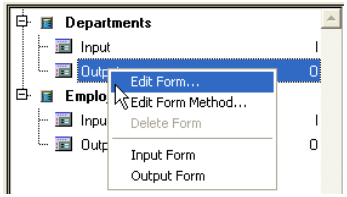
Creating a New Form

To create a new form, highlight the desired table and click **New**. The Form Wizard appears, ready for you to create the new form. For more information about using the Form Wizard, see "Creating a New Form" on page 235.

Note You can also create a new form by choosing **New Form** from the **Design** menu.

Editing a Form

To edit an existing form, expand the table containing the form, highlight the desired form and double-click, or highlight it and click **Edit**. You can also use the **Edit Form...** command of the contextual menu:



You edit forms using the Form editor. For information about editing forms, refer to chapters 4 and 5.

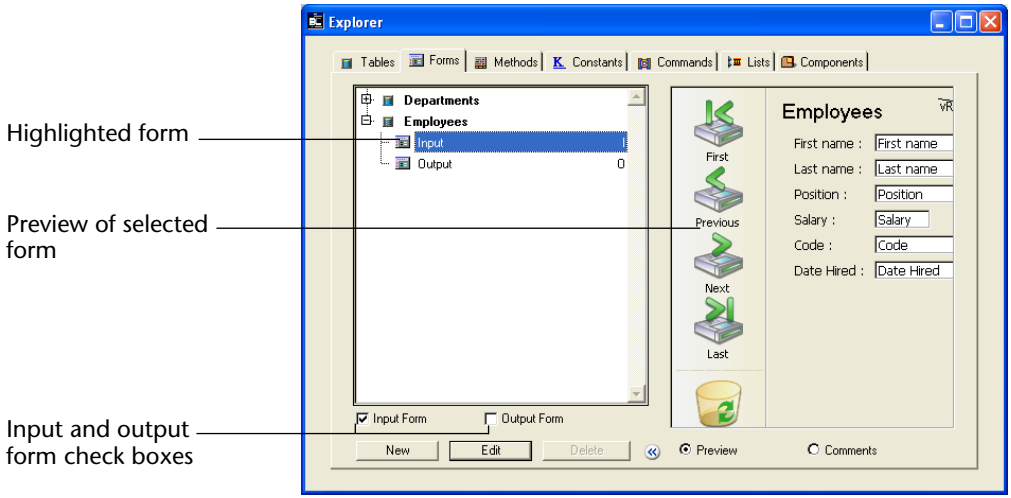
Deleting an Existing Form

To delete an existing form, select its name in the list and click on the **Delete** button or use the **Delete Form** command of the contextual menu.

Note You cannot delete a form if it is the current default input or output form for the table.

Previewing a Form

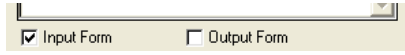
To display the preview area, click the icon . Highlight a Form in order to preview it.



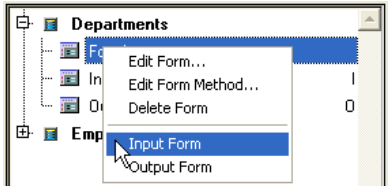
If you highlight a Table name in the Forms page, a preview of the table image from the Structure editor will appear in the preview area.

Designating a Form as the Current Input or Output Form

You can select the current input and output forms for each table using the Explorer. To do this, click the desired form name in the hierarchical list and then click either the **Input Form** or **Output Form** check box under the form list.



You can also click on the form and use the **Input Form** or **Output Form** commands of the contextual menu:



For more information on default input and output forms, see “Setting the Current Input and Output Forms” on page 262.

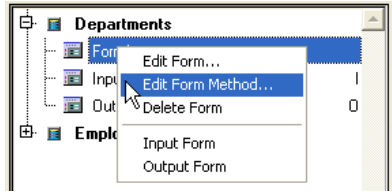
Using Drag and Drop

You can add a form name to a method by dragging. When you do so, the form name appears using the correct syntax. For example, if you drag the form “Input” in the [Companies] table, it will appear in a method as “[Companies];Input”.

You can add a subform to another form by dragging the name of the List form from the Forms page of the Explorer to the open form in the Form editor. You can add a Detail subform by holding down the **Shift** key and dragging the name of an input form from the Forms page of the Explorer to the subform area on the form. For more information, see the section “Adding a Subform to the Form” on page 468.

Modifying the form method

Using the contextual menu, you can open a form method directly from the Forms page of the Explorer:

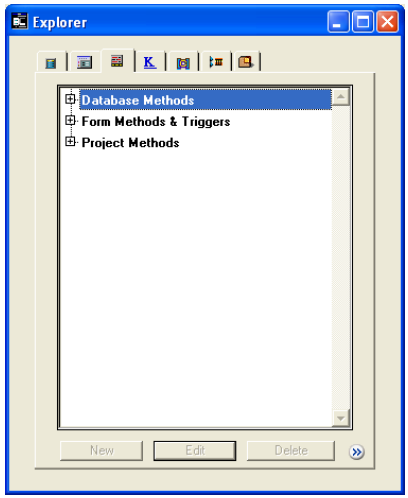


The form method will be opened in the Method editor window.

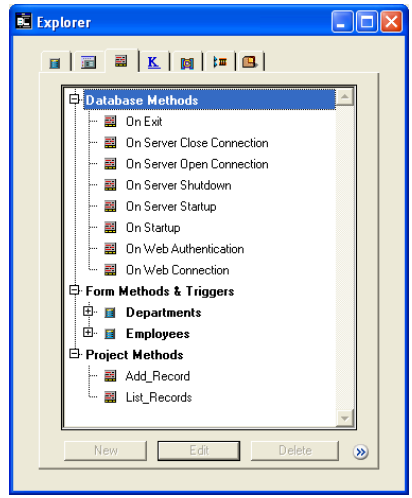
Methods Page

The Methods page lists the database, project, table (trigger), and form methods for the database. These types of methods are grouped into the categories: Database, Form/Triggers and Project.

Collapsed list



Expanded list



The Methods page is automatically selected when you select **Edit Method** from the **Design** menu.

Creating a New Method

Here is a description of how to create each type of method.

- **Project method** To create a new Project method, highlight the Project Methods item in the hierarchical list and click the **New** button.

Note You can also create a new project method by choosing **New Method** from the **Design** menu.

- **Trigger** To create a Trigger, expand the Form Methods & Triggers element in the hierarchical list and then highlight the desired table. Click the **Edit** button or double-click on the form name.

- **Form method** To create a Form method, expand the Form Methods & Triggers element in the hierarchical list, expand the desired table, and then highlight the desired form. Click the **Edit** button or double-click on the form name.

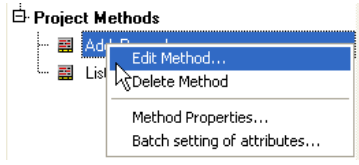
Note You can also create a new form method as follows:
• by selecting **Edit Form Method...** in the contextual menu on the Forms page of the Explorer,
• by selecting **Form Method...** in the **Form** menu of the Form editor,
• by clicking on the **Edit...** button of the Property List for the Form editor when no object has been selected.

- **Database method** You cannot create new Database methods. Instead, you can add code to an existing blank Database methods. To do so, expand the Database method item and double-click the desired Database method or highlight it and click the **Edit** button.

Modifying a Method

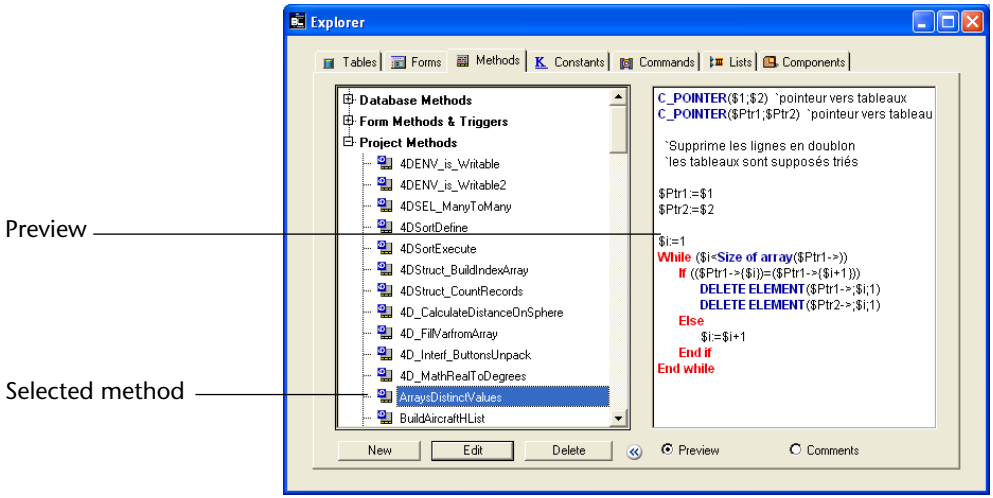
To modify an existing method, double-click its name, or highlight it and click **Edit**, or double-click its name in the preview area (when it is expanded, see below).

You can also choose the **Edit Method...** command in the contextual menu:



Previewing a Method

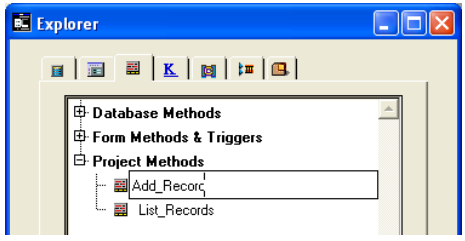
Click the icon to display the preview area and highlight the method you want to preview.



Once the preview area is open, you can preview other methods by highlighting them in the hierarchical list.

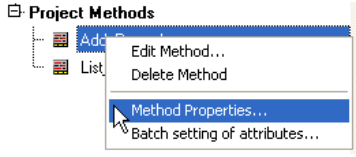
Renaming a project method or modifying its properties

Hold down the **Command** key (MacOS) or the **Ctrl** key (Windows) and click a method name. The text becomes editable.

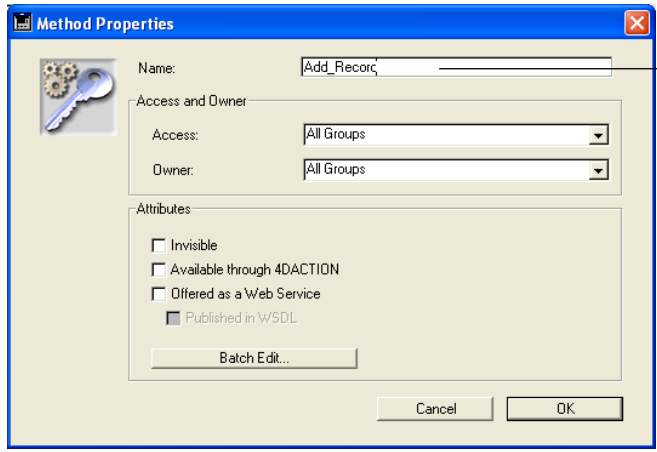


Enter the new name and click anywhere outside the entry area to save it. 4th Dimension then resorts the list of methods alphabetically.

You can also rename the method in the Method Properties dialog box. To do this, choose the **Method Properties...** command in the contextual menu:



The “Method Properties” dialog box appears, allowing you to modify the name as well as other properties of the method.



For more information about the other options of this dialog box, refer to the paragraph “Defining the properties of project methods,” page 541.

Batch setting of attributes

The **Batch setting of attributes...** command of the contextual menu (applied to project methods only) is used to modify an attribute (Invisible, Available through 4DACTION, etc.) for all or part of the database project methods in a single operation.

For more information about this command, refer to the paragraph “Batch Setting for Method Attributes,” page 545.

Using Drag and Drop

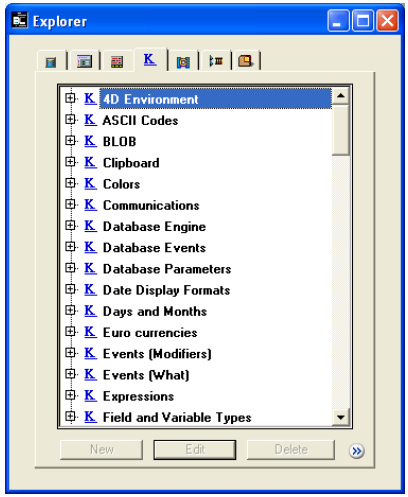
When you are writing a method, you can add the name of another project method using drag and drop. Highlight the desired method name in the Explorer and drag it to the Method editor window.

When creating custom menus, you need to assign a project method to each menu command. You can do so by dragging a project method name from the Explorer to the Menu Bar editor. For more information, see the section “Assigning Methods to Menu Commands” on page 598.

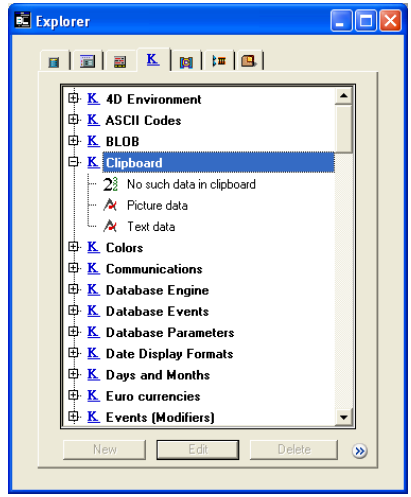
Constants Page

The Constants page contains a hierarchical list of all the constants that can be used in methods.

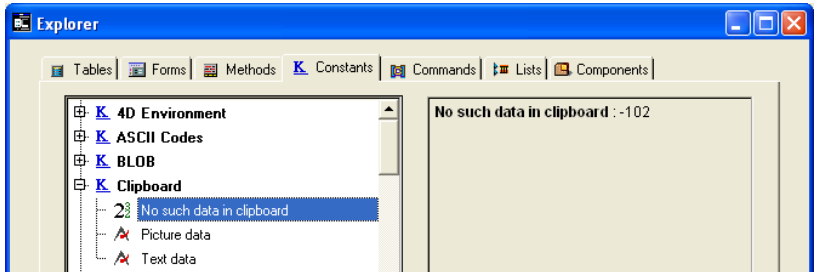
Collapsed list



Expanded list



If the preview area is displayed, it shows the value of the highlighted constant.



The constants displayed may come from 4th Dimension, plug-ins or customized resources. For information on using constants in methods, see the section "Constants" in the *4th Dimension Language Reference* manual.

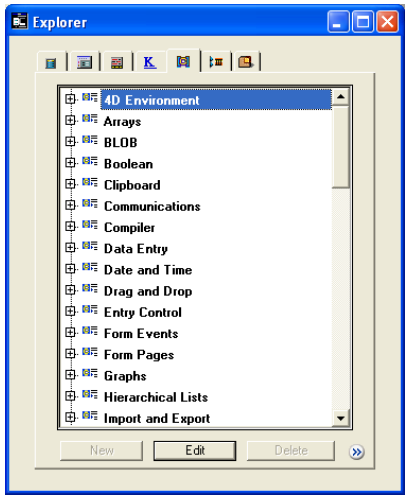
Using Drag and Drop

You frequently use constants in your methods. Instead of typing the constant you can add a constant to a method from the Explorer. Highlight the desired constant and drag it to the Method editor window. By default, the constants will be underlined when the Method editor parses the line of code.

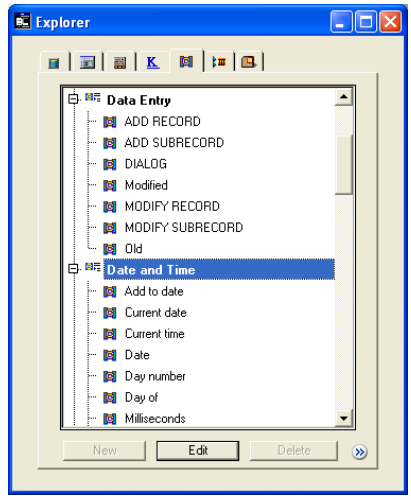
Commands Page

The Commands page displays all built-in 4th Dimension commands, grouped by category. It is equivalent to the list of commands shown in the bottom-right scrollable area of the Method editor.

Collapsed list



Expanded list

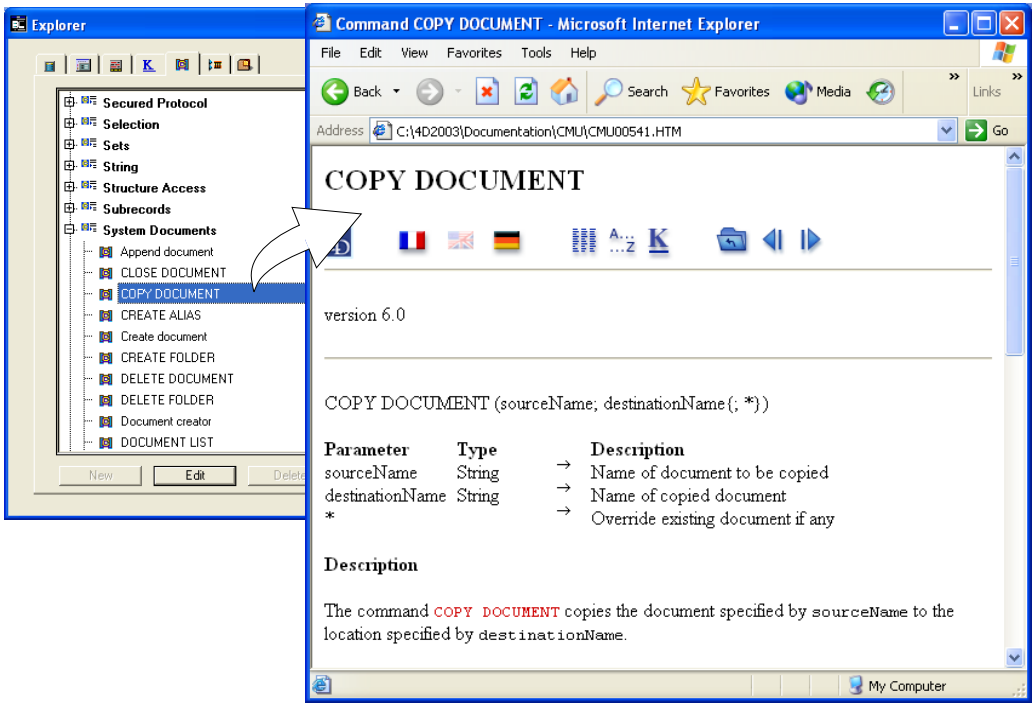


You can use this page to access on-line documentation for the commands. The preview area also allows you to display either the documentation (only under MacOS) or information about the command syntax (Windows).

Access to the on-line documentation

You can now access the on-line (HTML) documentation of 4th Dimension directly from the Commands page: to do this, double-click on the name of the command or select it and click on the **Edit** button.

The corresponding HTML page will be displayed in your browser.

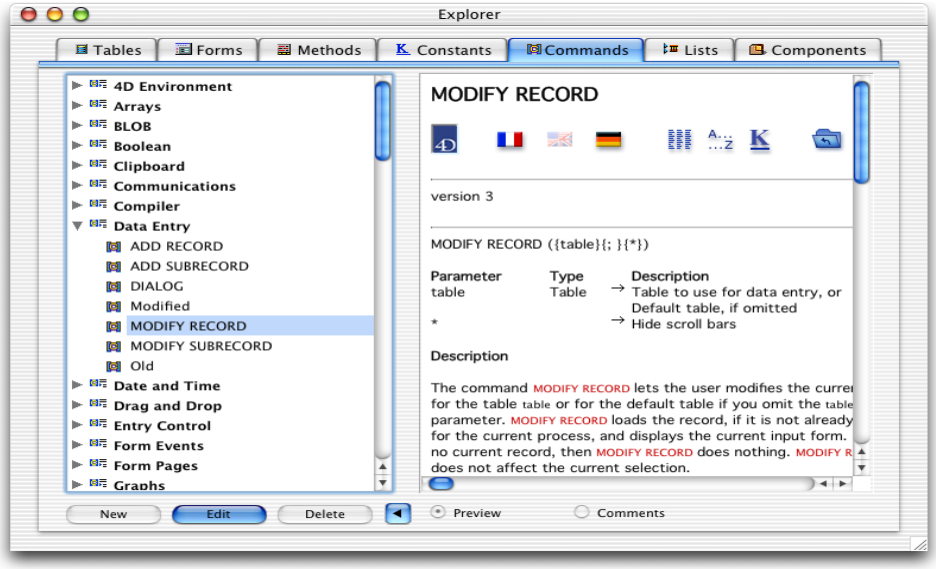


If your browser is already open, this page replaces the previously displayed one. If no browser is open, 4th Dimension launches the default browser before displaying the page.

HTML documentation pages can be stored on a CD-Rom, on your hard disk, or can come directly from the 4D, Inc. Web site. The location from which the page is loaded is defined by the “Documentation Access from the Explorer” parameter in the application Preferences (see the paragraph “Documentation Page,” page 129).

Displaying documentation in preview area (MacOS only)

Under MacOS, when you select a 4D command in the Explorer, its HTML description is displayed in the preview area:

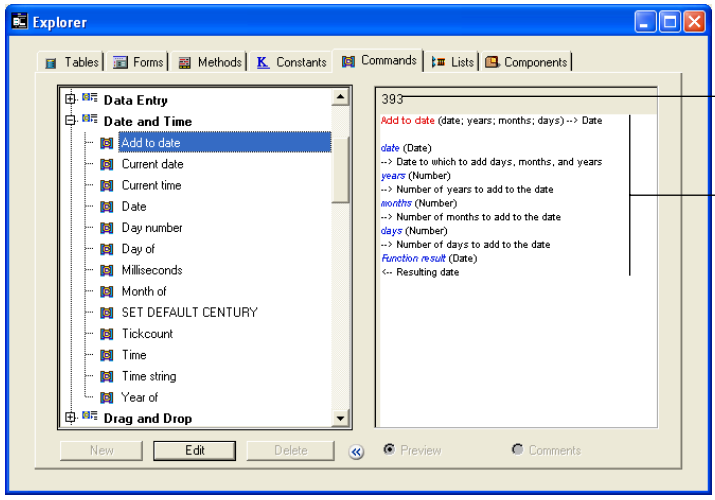


The navigation buttons and hypertext links are active on each displayed page. On the other hand, it is not possible to use the standard Web navigation functions (Next / Back), nor to copy information.

The location from which the page is loaded depends on the parameters defined in the application Preferences (see the paragraph “Documentation Page,” page 129).

Displaying the command syntax (Windows)

When the preview area is open, it displays the number of the selected command and a description of the command syntax:

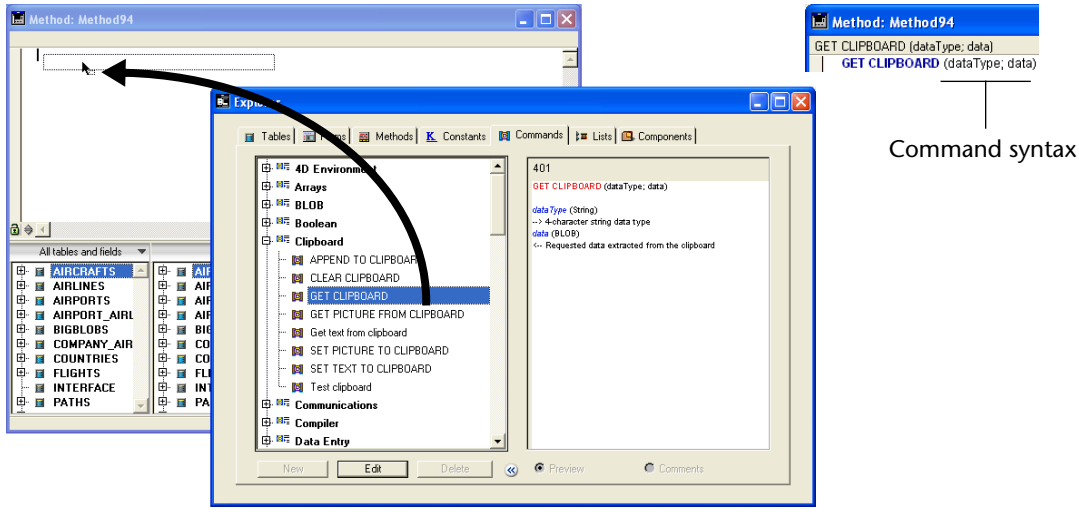


Note The command number is used by the Command name function (for more information, refer to the *Language Reference* manual of 4th Dimension).

If you do not want to display the syntax, click on the command name while holding down the **Alt** key.

Using Drag and Drop

Using drag and drop, you can add a command to a method. Highlight the desired command and drag it to the Method editor window. By default, the command and its syntax are inserted at the same time:



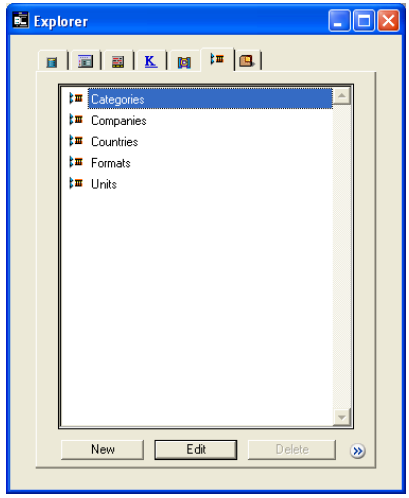
If you do not want to insert the syntax, press the **Alt** key (Windows) or **Option** key (MacOS) when dragging the command.

Note The system that is used to pre-insert the command syntax (as well as to view it in the Explorer under Windows) uses a specific help file. To take advantage of this feature you should make sure this file is neither renamed nor moved.

- Under MacOS, the file is named 4D Help and should be located in the active 4D folder of your system or in the 4D application folder.
- Under Windows, the file is named 4D Help.rsr and should be located in the active 4D folder of your system or in the 4D application folder.

Lists Page

The Lists page lists all the lists that you have created using the Lists editor. There is no expanded view of lists.



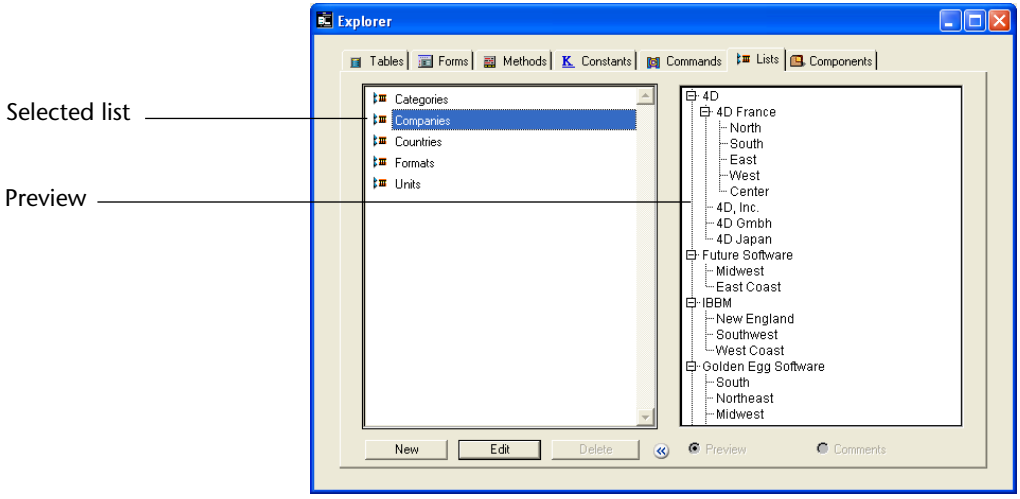
Creating or modifying a list

To modify a list, click the name of the list then click on the **Edit** button, or double-click it. You can also double-click in the preview area.

To create a list, click **New**. The List editor then appears. For more information on how to modify or create a list, refer to Chapter 11, “Creating Lists” on page 641.

Preview of a list

To display the preview area of the Explorer, click on the preview icon . Select a list in order to display it in the preview area.



Hierarchical lists will be displayed expanded.

Using Drag and Drop

You can create a hierarchical list or a drop-down menu in the Form editor by dragging a list from the Explorer window into the Form editor. The list is then automatically assigned to the object created in the Form editor.

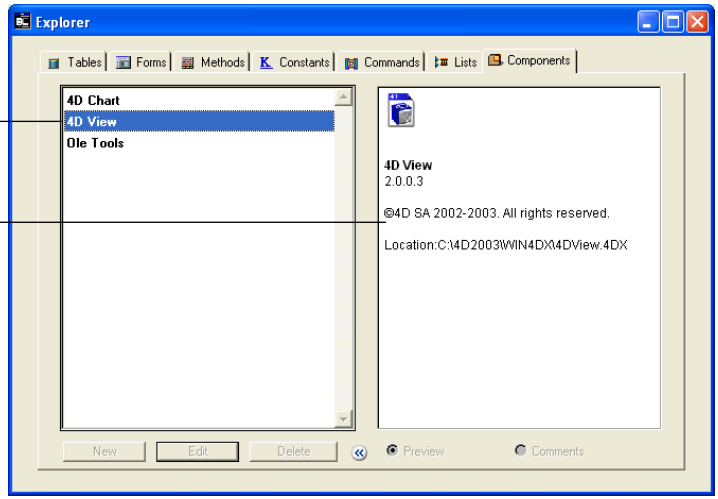
- To create a hierarchical list, drag a list from the Explorer window into the Form editor window.
- To create a drop-down menu, drag a list from the Explorer window into the Form editor window while holding down the **Shift** key.

Components and Plug-ins Page

The Components page lists all 4D components and plug-ins installed within a database as well as the properties of a selected component.

List of components and plug-ins installed in the database

Properties of the selected element (provided by the developer)



The component properties appear in the preview area of the Explorer, once it has been expanded: Component name, Version, Creation date and Comments. This information, provided by the component developer, allows precise component identification.
The plug-in properties include the location of the plug-in files.

The Components page allows you to view only components installed in the database. Installation and management of components is handled through the intermediary of 4D Insider.

About 4D components

A component gathers a set of 4D design objects (tables, methods, forms, menu bars, etc.) that represent one or more additional functions. For example, you can develop an e-mail component using 4D.

Components integrate the idea of object "protection": each object within a component is labelled as "public," "protected" or "private," which determines if it will be visible or able to be modified once the component is installed. Components allow 4D developers to broadcast original solutions in total security.

Component object attributes

Once a component has been created using 4D Insider, each object receives one of the following three attributes: **Public**, **Protected** of **Private**. These attributes, at the base of the component protection system, indicate if the concerned objects will be visible or able to be modified in 4th Dimension Design environment and in 4D Insider, once the component has been generated and installed.

- **Public**: "public" objects are visible and able to be modified by users, however, they can neither be renamed or deleted. This type of object can be useful in letting users customize objects. In 4D editors, public objects appear like all other objects.
- **Protected**: "protected" objects are visible but can neither be modified nor deleted by users. A protected method can be called, but its content can neither be displayed nor modified (the preview are in Explorer remains empty).
  In 4D editors, icons for protected objects have a red slash through them:

   MyMod_Prot_CallsTable1

- **Private**: "private" objects are neither visible nor, consequently, able to be modified by component users in both 4th Dimension and 4D Insider.

For a detailed description of components, please refer to the 4D Insider Reference guide.

Using Comments

The Explorer allows you to write comments about objects in your database. Using comments is particularly appropriate for databases being developed by multiple programmers and is good programming practice generally. Your comments are displayed in the preview area in the Explorer.

The following objects accept comments:

- methods (database methods, project methods, form methods and triggers),
- forms,
- tables and subtables,
- fields.

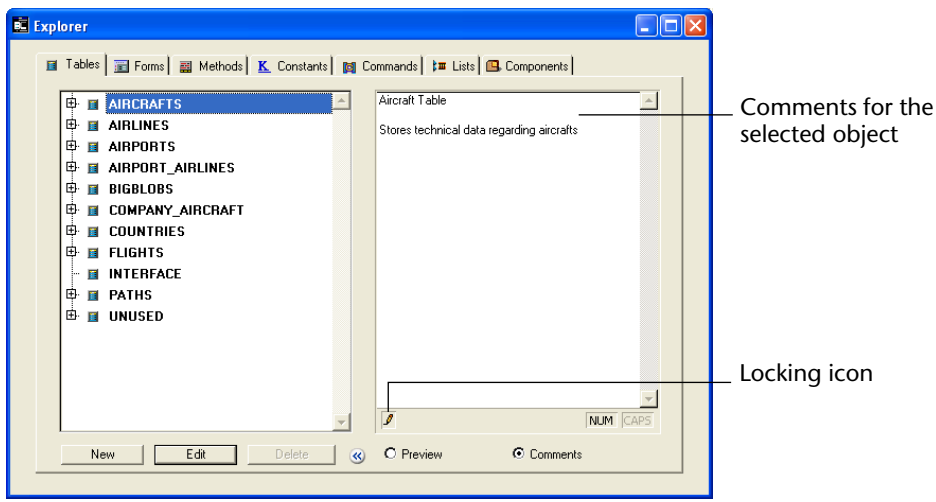
A comment can be entered as styled text (i.e., the characters can have different font styles or colors, etc.) that can be modified and viewed at any time in Design environment. It can contain a description of an object to which it is associated as well as any information necessary to understand how the object functions in the database. The comments you create are stored in the database's structure.

Moreover, 4th Dimension allows you to generate automatic comments, which means that 4th Dimension automatically enters comments when an object is created or modified.

Note Comments generated by 4th Dimension are compatible with 4D Insider's comments.

Associating a Comment to an Object

You create, view, and modify comments from the Explorer. To access the Comments area for an object, select the object and then click the Comments button located above the preview area.
When the **Comments** option is selected, the preview area is replaced by the Comments area.



You can enter up to 32 KB (32,700 characters) of text for each object. The Comments area has a vertical scrollbar that allows you to scroll through the text.

4D Server The Locking icon, located in the bottom left corner of the area, indicates if the comment is already being edited by another user. If this is the case, the pencil has a slash through it and the comment can only be viewed.

Note The NUM and CAPS buttons to the right of the Locking icon indicate whether the NUM or CAPS Lock functions on your keyboard are engaged.

► To view, create, or modify a comment:

1 **Select the object (method, table, or field) to document in the left portion of the Explorer.**

2 **Click the Comments radio button, if it has not already been selected.**
The preview area changes to an editable text entry area.

3 **Enter or modify the text in the comments area.**
The text is saved as soon as you click outside of the entry area. You can use the standard text editor commands (**Copy**, **Paste**, **Select All**, etc.) available in the **Edit** menu or by using keyboard shortcuts in the Comments area. You can also navigate the text in the comments area by using keyboard shortcuts as you would for any other text area.

Modifying the Style of the Comments

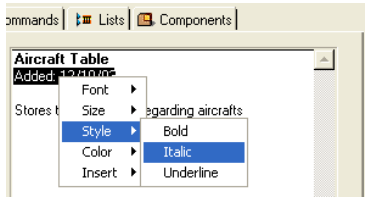
You can enhance the style of the comments (add bold or italics) and change the font attributes (font, font style, font size, or color) using a contextual menu.

► To modify the style of the comments' text:

1 **In the Comments area, select the text that you want to modify.**

2 **Under Windows: Click in the area with the right mouse button.**
or
Under MacOS: Control+click in the area.

A hierarchical pop-up menu appears:



3 Select the attributes that you want to apply to the text.

Inserting the Date, Time, or User Name

You can insert the date, time and user name (as defined in the 4th Dimension Password editor) into your comments.

Note These shortcuts are particularly appropriate for the automatic generation of comments (see below), but can also be used for comments about an individual object.

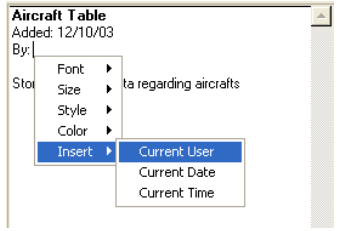
▶ To insert the date, time, or user name in a comment:

1 In the Comments area, place the insertion point where you want to insert the information.

2 Under Windows: click in the area with the right mouse button.
Or
Under MacOS: Control+click in the area.

A hierarchical pop-up menu appears:

3 Select the value you want to insert from the Insert menu.



The selected information is inserted immediately into the Comments area with its current value. If the database doesn't have a password system, the **Current User** command displays "Designer."

Note This shortcut helps you enter data, but does not allow you to insert variables. Information inserted in this way can only be updated manually. To insert automatic comments, please refer to the following section.

Inserting Automatic Comments

You can activate an automatic commenting system that can only be used for methods and forms in the database.

When this system is activated, a comment is automatically associated with every method or form created or modified in the database. An automatic comment can consist of both static text (such as "Modified by") and variables (such as the current date, current time, and user name).

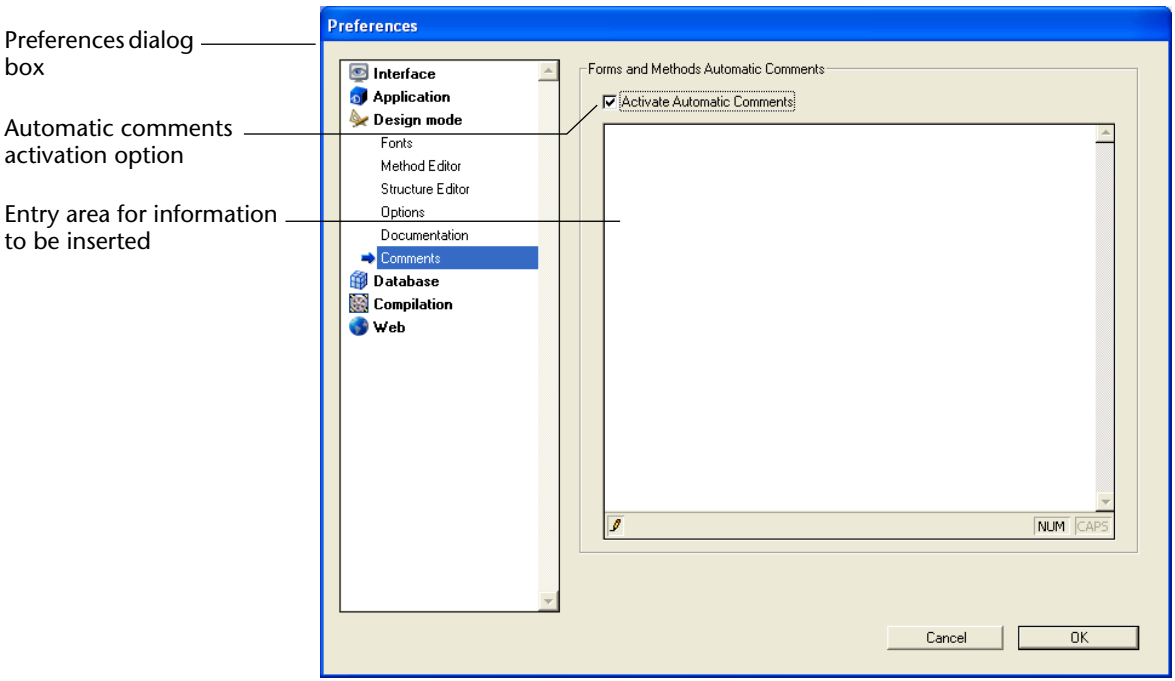
Automatic comments are defined in the Preferences dialog box of the application. As with standard comments, you can view them using the Explorer.

► To activate the automatic commenting system:

1 **Select the Preferences command in the Edit menu.**

2 **In the Preferences dialog box, display the Comments page of the "Design Mode" theme.**

3 **Select the "Activate Automatic Comments" option:**



4 **Enter the set text of the comments.**

5 **Insert the variable elements of the comments.**

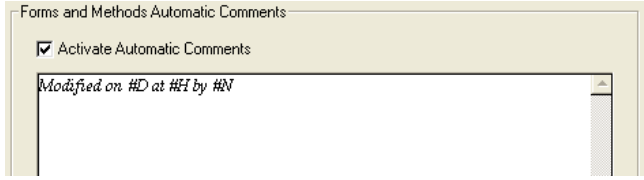
You can use the hierarchical pop-up menu (under Windows, click in the area with the **right button** of the mouse; under MacOS, use **Control+click**).

You can also directly enter variables that will be updated before being inserted in the comments:

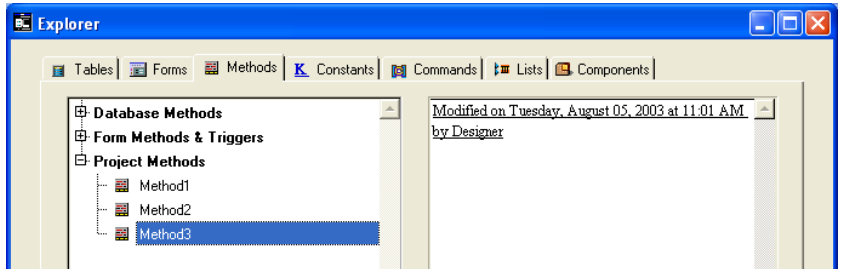
- **#D** for the date
- **#H** for the time
- **#N** for the current user

Note If your database doesn't have a password system, #N returns the string Designer.

For example, if you enter the following values:



The automatic comment will be added to the comments for all new methods and forms as well as all existing methods and forms that are modified after automatic commenting is turned on:



4D Server Automatic comments can be modified by any client workstation that has access to the Preferences. You can also modify these parameters in the Preferences of the server machine.

Every modification made to the Automatic Comments page is immediately taken into consideration by each client workstation as soon as an object is modified and its comments are accepted.

Runtime Explorer

The Runtime Explorer window allows you to view the behavior of the different structural elements in your database and to verify that the available resources are operating as expected. The Runtime Explorer is particularly useful in your database's development and analysis phase. For that reason, you will want to familiarize yourself with the 4th Dimension language and the process of developing a customized application before learning about the Runtime Explorer.

Displaying the Window

The Runtime Explorer window is accessible in all of 4th Dimension's environments: Design, User, and Custom Menus (compiled or interpreted).

Note Only the Designer and Administrator have access to the Runtime Explorer window.

The Runtime Explorer can be displayed in two types of windows: in a standard window (in the Design environment only) or in a floating palette (in all environments).The floating palette always remains in front of other open windows. The standard window behaves as any other Design environment editor window.

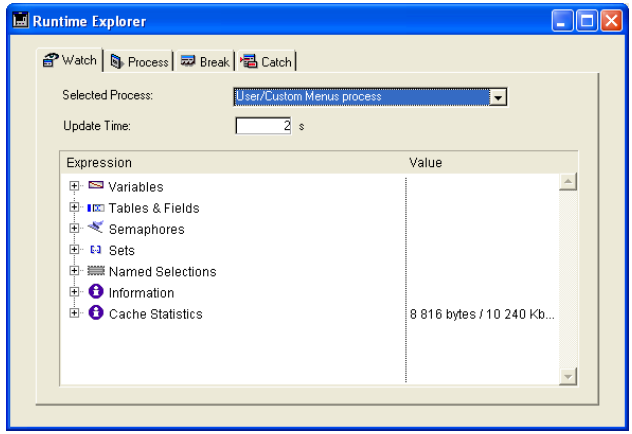
► To display the Runtime Explorer in a standard window (in the Design environment):

1 Choose Runtime Explorer from the Tools menu.

► To display the Runtime Explorer as a floating palette (from the Design, User, or Custom Menus environments):

1 Under Windows, press Ctrl+Shift+F9.
Under MacOS, press Command+Shift+F9.
or
Hold down the Shift key and choose Runtime Explorer from the Tools menu (Design environment only).

The Runtime Explorer window has four pages that are accessed by clicking on the following tabs: **Watch**, **Process**, **Break,** and **Catch**.



Watch Page

The **Watch** page is a debugger and displays information about code execution.

Note The information displayed in this page is identical to the information in the debugger's Watch pane. For more information about the Watch pane, refer to the "Debugger" section in the *4th Dimension Language Reference* manual.

- **Selected Process**: this drop-down list contains all the processes that are being executed in the database. It allows you to select the process(es) that you want to observe.
- **Update Time**: in this area, you can define a value (in seconds) that indicates how often the information in the page will be updated.

The Expression column displays the names of the objects and expressions. The Value column displays the current value of the objects and expressions. These columns can be resized, one in relation to another. To do so, click on the separation line and drag it to the left or right.

By clicking on a value in the right column, you can modify the object's value, if the object allows for this.

The multi-level hierarchical list is organized by theme. The themes are as follows:

- **Variables**: allows you to view the list of the database's interprocess variables as well as the list of the selected process's process variables.

- **Tables and fields**, **Semaphores**, **Sets**, **Named Selections:** the information provided in these themes is identical to the information provided by 4th Dimension's debugger. For more information, refer to the "Debugger" chapter in the *4th Dimension Language Reference* manual.

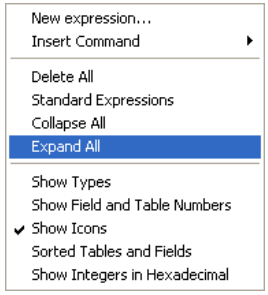
- **Information**: this theme displays general information, such as the current default table (if one exists), available memory, query destination, etc.
  The bottom information is only available if the Web server is active: **Web file to send**, **Web cache occupation** (number of pages present in Web cache as well as its use percentage), **Web server activity time** (duration of Web server use in hours:minutes:seconds format), **Number of HTTP requests** (total number of HTTP requests received since Web server launch, as well as the instantaneous number of requests per second). The expressions contained within this theme cannot be modified.

- **Cache statistics:** allows you to obtain information about 4th Dimension's cache. This list is also available in 4th Dimension's debugger. For more information about 4th Dimension's cache, refer to the paragraph "System Resources Page," page 115.

  To delete an expression or a theme, select the corresponding line and press the **Delete** key.

  You can also add a **New Expression** or a 4th Dimension **Command**, or perform global actions: **Delete All**, display all the **Standard Expressions**, **Collapse All** or **Expand All**.

  To do this, select the corresponding command in the contextual menu which appears when you click the **Right mouse button** (Windows) or **Control+click** (MacOS) in the window :



In addition, several display options are available in the lower part of the contextual menu.

- **Show Types**: displays or hides the types of fields next to their names in the list of tables & fields.
- **Show Field and Table Numbers**: displays or hides the table and field numbers next to their names in the list of tables & fields. For each field, the following format is applied: [TableNum]FieldNum.
- **Show Icons**: displays or hides the object icons in the hierarchical list.
- **Sorted Tables and Fields**: sorts the list of tables and fields by alphabetical order (by default, these objects appear in the order that they were created).
- **Show Integers in Hexadecimal**: displays the variables declared as Integer or Long Integer types in their hexadecimal form.

Process Page

The **Process** page allows you to graphically view the CPU time consumed by each process as well as the state of each process.

You can show or hide the graphical display of each process by clicking on the expansion icon to the left of the process' name. You can also show or hide all the icons by clicking the **Show** and **Hide** buttons in the window's tool bar.

The use of the process page is described in Chapter 13, "Managing Processes" on page 671.

Break and Catch Pages

The **Break** page allows you to view and manage break points that you have placed in your code.

The **Catch** page displays the break points defined in the database in relation to commands (or expressions).

The general functioning of these pages is identical to the debugger's **Break List**. For more information, refer to the "Debugger" chapter in *4th Dimension Language Reference* manual. The commands in the **Break List** menu are replaced by the buttons in the window's tool bar.

Note You can also define break points directly in the Method editor (refer to "Using the Method Editor" on page 548).

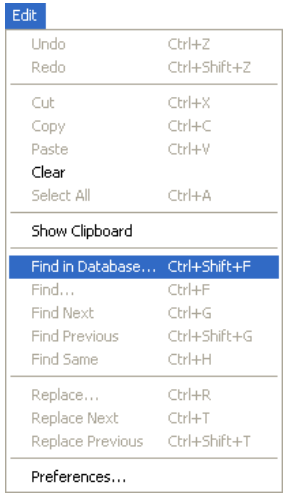
Searching in the Database

The Find dialog box allows you to search for a string in part of or in the entire database structure. You can, for example, search for the string "MyVar" only in methods. The search can go as far as searching objects in all the database's forms.

Find window

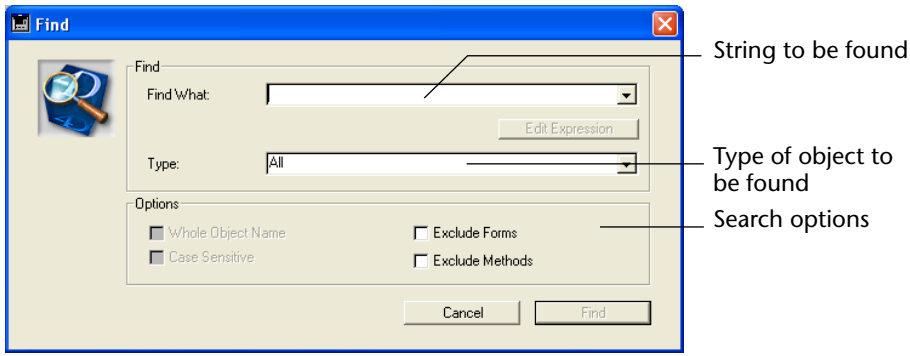
► To carry out a search in the database structure:

1 Choose the Find in Database... command from the Edit menu.



Note The other find commands shown in this menu are intended for search and replaces in the Method editor. For more information about this, refer to the paragraph "Find and replace," page 582.

The general find window appears:



2 **In the "Find What:" area, enter a string to find.**

Note The at-sign (@) is considered a character. It is not possible to use it as the wildcard character in the Design environment Find dialog box.

3 **If desired, specify the object type to which you want to restrict the search.**

- Restricting the search to a certain object type makes the search faster. If you don't want the search to be limited to an object type, choose **All** in the **Type** menu.
- If you select **Expression** in the **Type** menu, the **Edit Expression** button becomes enabled. It allows you to define the expression to search for directly in the Formula Editor.

The object types are described in the paragraph "String Types and Scope of the Search," page 96.

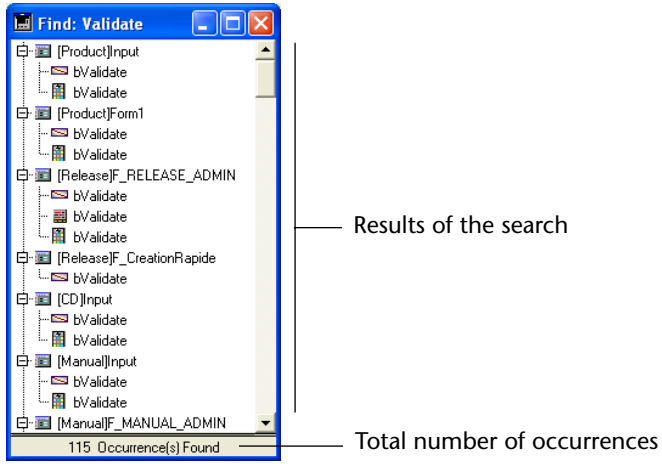
4 **Define (if necessary) the search options.**

These options are described in the paragraph "Searching Options," page 97.

5 **Click the Find button or press the Enter key.**

The search begins.

Once the search has finished, the occurrences found appear in a new window and are presented in a resizable hierarchical list organized by type of object found (tables and fields, forms, methods, menu bars, lists, pictures).



You can double-click on a line in this window to view the object in its editor.

If you do several searches, each search opens its own result window, leaving previous result windows open.

Once you have executed a search, the value entered in the Find What area is saved in memory. This value, as well as all the other values entered during the same session, can be selected from the combo box:

Using the combo box, you can quickly do the same search many times.

String Types and Scope of the Search

The Find dialog box allows you to limit the search to a particular object type or search across all object types:

- An expression, for example “total:=Sum([Accounts]Total)”
- A variable, for example “$vpPicture1”
- A table or field name, for example “[Clients]Name”
- A form object name, for example “Background”
- A comment, for example “Modified on”

You can also search among all object types.

By default, the search will be done throughout the entire structure of the database. You can, however, exclude methods and/or forms from the search.

Depending on the designated object type, the search will be done among the following types of objects:

- Forms (which can also be excluded)
- Methods (which can also be excluded)
- Menus and menu commands in custom menus
- Lists
- Tables and fields (as well as subtables and subfields)
- Comments
- Pictures from the Picture library.

The following table presents the structure elements in which you can search, depending on the different object types:

	Places in which to search						
Object types	Forms and form names	Methods and method names	Menus/menu commands	Lists	Tables and fields (Structure editor)	Comments	Picture library
Expression		X	X				
Variable	X	X					
Table or field name	X	X			X		
Form object name	X	X					X
Comment						X	
All	X	X	X	X	X	X	X

Searching Options

You can select different options presented to you as check boxes. Depending on the type of search you specify, some options may be disabled:

- **Whole Object Name**
  When this option is selected, the search is limited to the exact occurrences of the searched object name or expression. In this case, for example, if you are searching for "client," 4th Dimension will not find "clients" or "myclient."

  By default, the option is not selected, which means that searching for "var" will find "Myvar," "variation," etc.

- **Case Sensitive**
  This option is selectable only if the option **Whole Object Name** is selected. When **Case Sensitive** is selected, the search takes into account the case of the characters as they have been entered in the Find dialog box. Therefore, if you search for "MyVar," 4th Dimension won't find "myVar."

- **Exclude Forms**
  When this option is selected, the search is done throughout the database, except in forms and form names.

- **Exclude Methods**
  When this option is selected, the search is done throughout the database, except in methods and method names.

Platform Interface

The Platform Interface property lets you display any form using the conventions of the graphical user interface (GUI) of a selected platform. Specifying a Platform Interface for a form or object does not actually modify the form or object. The selected platform interface affects only the way the forms are displayed on the screen. A form can look like a Windows 95/98/2000/NT 4, Windows 3.11/NT 3.51, MacOS 7, MacOS 9 or Mac Theme screen depending on the option chosen.

The properties of the platform only affect the forms of 4th Dimension. They do not affect the standard dialog boxes of 4th Dimension such as the Query or Order by editors. 4th Dimension always displays the dialog boxes that correspond to the graphical user interface (GUI) of the current platform.

Where to set the platform

You can set the platform interface at three different levels. By default, each level "inherits" the property of the next highest level.

- At the **database** level: the platform interface is set in the Preferences. In this case, the interface set is applied by default to all the forms of the database.
- At the individual **form** level: the platform interface is set in the Form Properties window. In this case, the interface set at the form level replaces that of the database.
  You can set the platform interface of a form when you create it, or you can modify it using the Form editor.
- At the individual **form object** level: the platform interface is set in the Object properties. In this case, the interface set at the object level replaces that of the form and/or database.

  For information on setting the platform interface at the database level, see the paragraph "Look Page," page 110.
  For information on setting the platform interface when the form is being created, see the paragraph "Setting the Appearance of Form Objects," page 258. For information on modifying the platform interface settings of a form using the Form editor, see the paragraph "Setting the Platform," page 292.
  For information on changing the platform interface of individual objects, see the paragraph "Platform Interface and Border Line Style," page 335.

Platform Interface Settings

At the general Preferences level, the **Automatic** option is the default option: the appearance of the forms is therefore related to the host platform. The Automatic option can be defined using two pop-up menus: the **Best Match** option corresponds to the most "modern" interface in relation to the capabilities of the machine where the program is run:

- Under Windows, the **Windows 95/98/2000, NT 4** interface is used regardless of the version of Windows.
- Under MacOS, the **Mac Theme** interface is used unless the application is run from a machine that is too old to support it, in which case the **Mac OS 9** interface is used.

Choosing another option in the Platform menu or in the Automatic option menus will allow you to handle the following situations:

- You are designing forms on one platform and you would like to see how they will look on the other platforms or operating systems.
- No matter which platform you are using, you want your layouts to always look the same, overriding the default GUI of the platform.
- You would like to control the platform interface using the language. The command SET PLATFORM INTERFACE allows you to choose the platform interface. With SET PLATFORM INTERFACE, you can simply let the user(s) of the database choose the GUI they like.

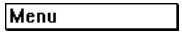
This setting acts on the following objects and form properties:

- Fields
- Buttons (push buttons
- Thermometers and dials
- Check boxes
- Radio buttons
- Hierarchical lists
- Pop-up menus and drop-down lists
- Tab controls
- Objects whose foreground and/or background colors are automatic
- Form background color.

Here are descriptions of each setting:

MacOS 7

- Buttons are displayed in MacOS 7 format.
- Check boxes and radio buttons are displayed as MacOS 7 controls.
- Thermometers, rulers and dials are displayed as MacOS 7controls.
- Hierarchical lists are displayed according to MacOS 7 standards.
- Pop-up menus/drop-down lists are displayed according to MacOS 7 standards:



- The automatic foreground color of layout objects is set to black.
- The automatic background color of layout objects is set to white.
- Each form's background color is white.

Windows 3.11, NT 3.51

- Buttons are displayed as 3D effects push buttons that are characteristic of these versions of Windows.
- Checkboxes and radio buttons are displayed as regular Windows controls.
- Thermometers, rulers and dials are displayed as Windows controls.
- Hierarchical lists are displayed according to Windows standards.
- Pop-up menus/drop-down lists are displayed according to Windows standards:

- The automatic foreground color of layout objects is set to black.
- The automatic background color of layout objects is set to white.
- Each form's background is white.

Windows 95/98/2000, NT 4

With the Windows 95/98/2000, NT 4, interface, the buttons and graphic elements use the colors set in the Windows Control panel.

- Buttons are displayed as 3D effects push buttons that are characteristic of these versions of Windows.
- Check boxes and radio buttons are displayed as 3D effects like regular Windows controls.
- Thermometers, rulers and dials are displayed as Windows controls,
- Hierarchical lists are displayed in Windows format,
- Pop-up menu/drop-down lists are displayed in Windows format,

- The automatic foreground color is set by the user in the Windows Control panel.
- The automatic background color is set by the user in the Windows Control panel.
- Each form's background is set to the Button Face color chosen by the user in the Windows Control panel.

Note For more information about the automatic foreground and background colors of objects, see the paragraph "Foreground and Background Colors," page 344.

Mac OS 9

- Buttons are displayed in Mac OS 9 format.
- Check boxes and radio buttons are displayed as Mac OS 9 controls.
- Thermometers, rulers and dials are displayed as Mac OS 9 controls.
- Hierarchical lists are displayed according to the Mac OS 9 format.
- Pop-up menus/drop-down lists are displayed according to the Mac OS 9 format:

- The automatic foreground color is set to black.
- The automatic background color is set to gray.
- Each form's background color is gray.

Mac Theme

With the Mac Theme interface, the forms of 4D take the appearance of the theme defined in the "Appearance" control panel of MacOS.

- Buttons are displayed as defined in Mac Theme.
- Check boxes and radio buttons are displayed according to the Mac Theme set.
- Thermometers, rulers and dials are displayed according to the Mac Theme set.
- Hierarchical lists are displayed according to the Mac Theme set.
- Pop-up menus/drop-down lists are displayed according to the Mac Theme set,
- Under MacOS, tab controls can be right or left justified or even at the bottom (see the paragraph "Tab Controls," page 445),
- Under MacOS X, all objects benefit from Quartz rendering (based on anti-aliasing),

- Under MacOS X, the default buttons are pulsing buttons,

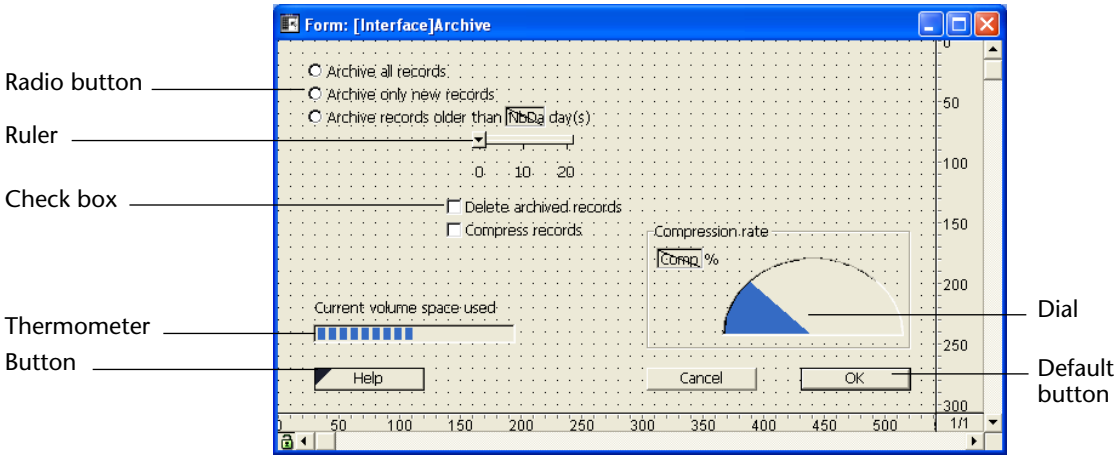
- The automatic foreground color objects depends on the Mac Theme set.
- The automatic background color depends on the Mac Theme set.
- Each form's background color depends on the Mac Theme set.

Under Windows, if you change colors in the Windows Color control panel while a 4th Dimension session is running, the program immediately updates the forms displayed in all running processes according to the new color settings. If you choose Windows 95/98/2000, NT 4 as the Platform Interface setting under MacOS, the automatic foreground and background colors are dark gray and light gray, respectively.
If you choose the Mac Theme setting for a form displayed under Windows, the standard Windows 95/98/2000, NT 4 appearance is displayed.

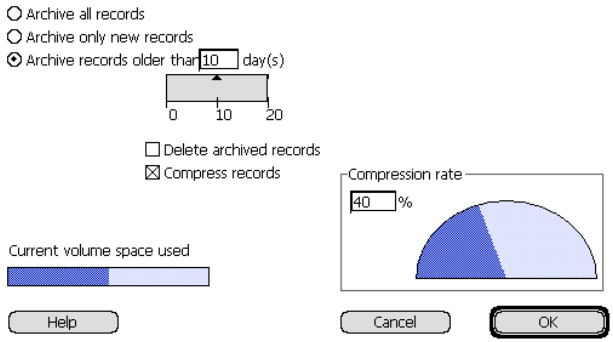
On any platform, changing the Platform Interface setting has an immediate effect, whatever the selection mode (by method or manually): all the forms and all the processes immediately reflect this change — if the form platform property is "Inherited from Database" and that of objects is "Inherited from Form."

4D Server Several 4D Client machines can concurrently use different Platform Interface settings.

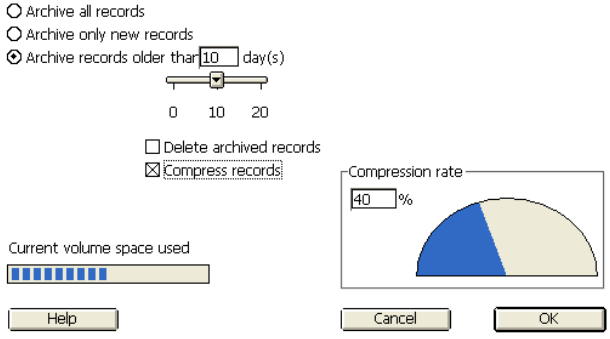
▼ This example illustrates the various appearances that a form and its objects can take according to the platform interface selected in the Preferences:



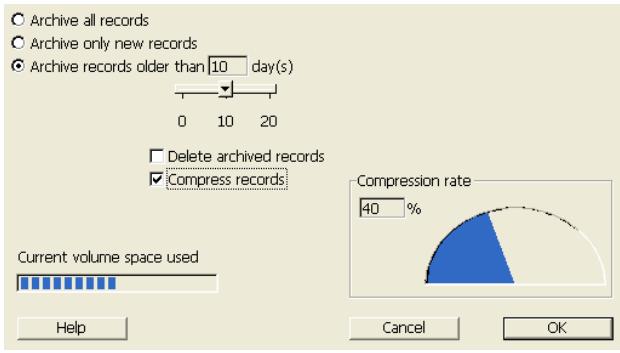
■ **MacOS 7 option**



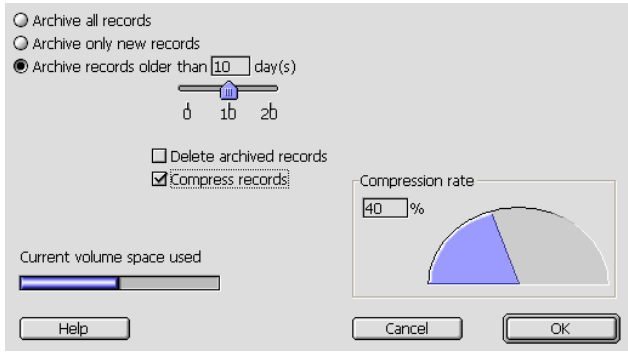
■ **Windows 3.11, NT 3.51 option**



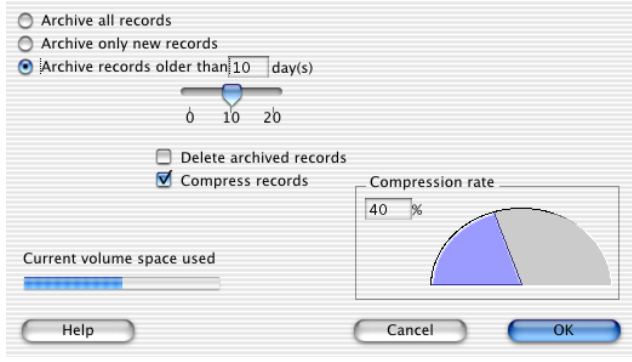
- **Windows 95/98/2000, NT 4 option**



- **Mac OS 9 option**



- **Mac Theme option** (varies depending on selected theme)



4D Server Considerations

4D Server supports the continuing development of a database that is currently in use by allowing you to modify the design of a database in the Design environment while users continue to work with the database in the User or Custom Menus environments. Changes to the design are visible to users as soon as they are saved (when you close the editor window or choose **Save *EditorName*** from the **File** menu).

In addition, 4D Server allows several users to work on the database design concurrently. 4D Server protects against users erasing each other's changes in the Design environment by providing an object-locking system in which structure objects are locked while they are being modified by a user. Structure objects include the following: field definitions, table definitions, forms, methods, menus, passwords, preferences, and lists. During the time that an object is locked, other users cannot modify it. An object is unlocked when it is closed.

For more information about how object locking works with a particular structure object, refer to the appropriate chapters in this manual.

2 Setting Preferences

This chapter describes each parameter of the Preferences dialog box of 4th Dimension.

Access to the Preferences

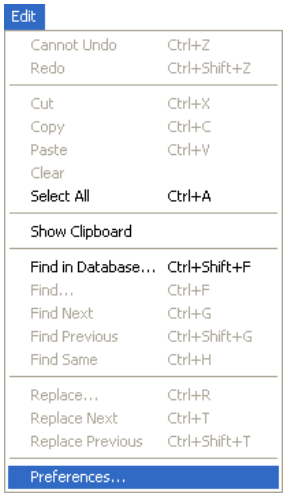
You can set application Preferences in the Design environment. Certain preferences will only apply to the open database while others will apply to all databases opened with your application.

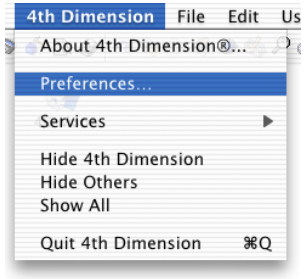
Most of the specifications that you select in the Preferences dialog box take immediate effect except for those that cannot take effect until the database is opened again (such as Startup environment).

4D Server Object locking occurs when two or more users try to modify the settings in the Preferences dialog box at the same time. Only one user can use the Preferences dialog box at a time. For more information about object locking, refer to the section "4D Server Considerations," page 105.

► To display the Preferences dialog box:

1 Under Windows and MacOS 9: choose Preferences from the Edit menu.
Under MacOS X: choose Preferences from the Application menu.





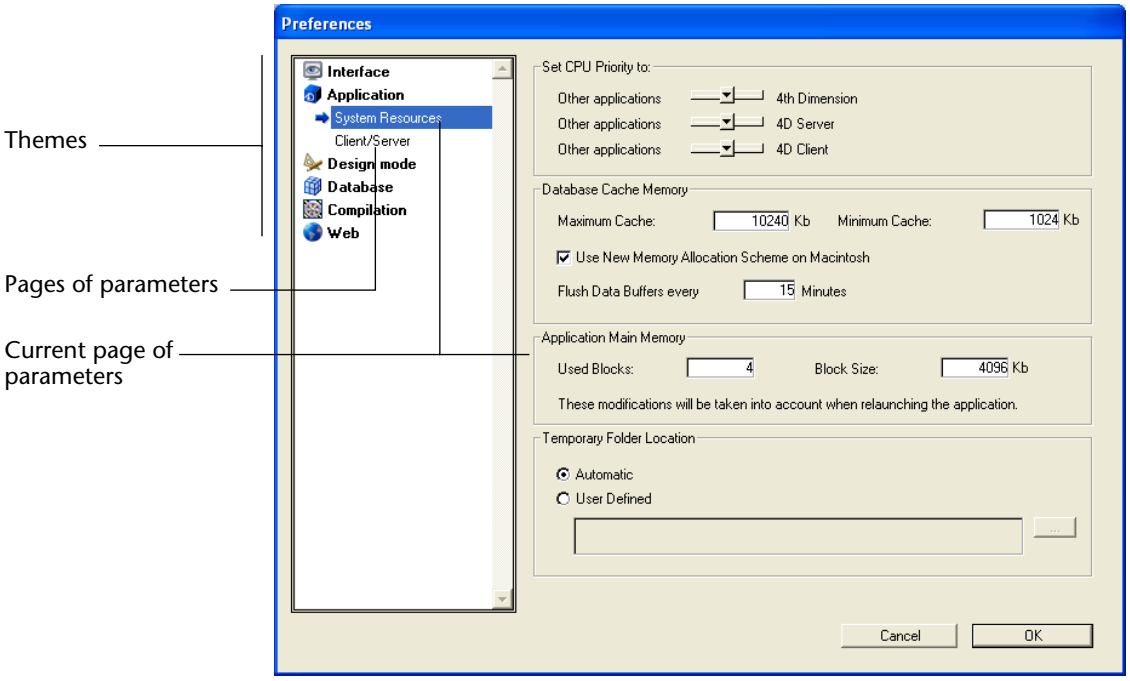
MacOS X

OR
Right-click (under Windows) or Control+click (under MacOS) in the background of the Structure editor, then select Preferences from the contextual menu.

The Preferences dialog box has 18 pages of parameters, grouped under 6 themes: **Interface**, **Application**, **Design mode**, **Database**, **Compilation** and **Web**.

To display a page of parameters in the right-hand part of the window, you just need to expand the corresponding theme and select the name of the page in the list on the left-hand side.

You can use the **arrow keys** of the keyboard to expand/contract and select the themes. The **Tab** key lets you select the values of the page of parameters sequentially.

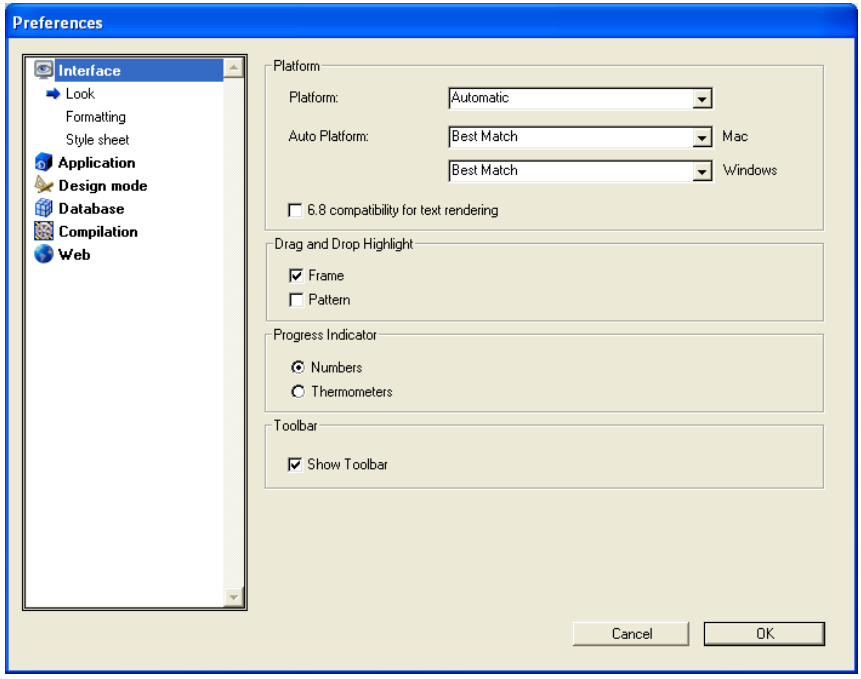


Interface Theme

The pages of this theme are used to set the properties of the appearance of your application on screen as well as the display formats and style sheets.

Look Page

The Look page lets you set the platform interface, the graphic representation of the drag-and-drop highlight, the progress indicator and whether or not to display the toolbar:



Platform

These options let you modify the appearance of the forms and form objects of your database.

The **Platform** menu includes six options:

- Automatic (default option),
- Mac OS 7,
- Windows 3.11, NT 3.51,
- Windows 95/98/2000, NT 4
- Mac OS 9
- Mac theme.

The **Auto Platform** menus (**Mac** and **Windows**) let you set the interface to be used, for each platform, when the **Automatic** option is selected. Each menu includes the **Best Match** option.

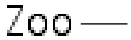
When this option is selected, 4D chooses the most “modern” interface as the automatic platform in relation to the capacities of the machine where the program is running:

- Under Windows, the **Windows 95/98/2000, NT 4** interface is used, regardless of the Windows version.
- Under MacOS, the **Mac Theme** interface is used unless the application is running from a machine that is too old to support it, in which case the **Mac OS 9** interface is used.

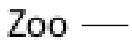
For more information about the platform interface, refer to the section “Platform Interface,” page 98.

- **6.8 compatibility for text rendering**: when it is checked, this option disables the “100% Quartz” display mode under MacOS X. 4th Dimension uses a Quartz graphic rendering engine to handle the display of all the text of your databases (including that of plug-ins) under MacOS. In versions 6.8.x of 4D, fields and variables of the text type, as well as plug-ins, did not benefit from Quartz rendering and were displayed using QuickDraw, the “conventional” rendering engine of MacOS.

  On the one hand, Quartz rendering causes anti-aliasing to be applied to the text to make its display more aesthetic and, on the other, it calculates character spacing more precisely.

  — “Conventional” display (QuickDraw)

  — Quartz display

  Since the modification of character spacing can alter application interfaces developed in versions 6.8.x, the “full” Quartz display mode can be disabled using the **6.8 compatibility for text rendering** option.

Notes

• Some specific text styles, such as “Shadow” and “Outline” attributes, are not supported by Quartz. The display of objects with these styles is therefore automatically handled by QuickDraw.
• The Quartz engine is only used for display purposes: printing is carried out in standard mode.
• For Quartz rendering to be visible, the database must be run on the MacOS X platform (version 10.2 minimum) and the Mac Theme platform interface must be used.

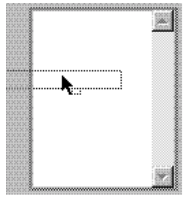
This option only has an effect under Mac OS X, but it can be selected with other platforms as part of the development of a multi-platform application.

This option is checked by default in applications created with previous versions of 4th Dimension. Conversely, it is unchecked by default in new databases created with 4th Dimension starting with version 2003.

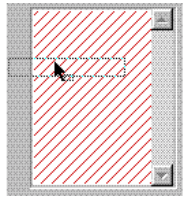
Drag and Drop Highlight

These options let you set the appearance of the drop area (the area that is receiving a dragged object). The Drop area takes on the appearance you specify when a dragged object is on top of it and it is "eligible" to receive the dragged object.

You can choose to highlight the area with a frame or a pattern (or both). These options apply to all the drag-and-drop operations in 4th Dimension —not only those of the Design environment. The following illustration compares the effects of these two options:



Frame



Pattern

Progress Indicator

This option selects the progress indicators: **Numbers** or **Thermometers**.

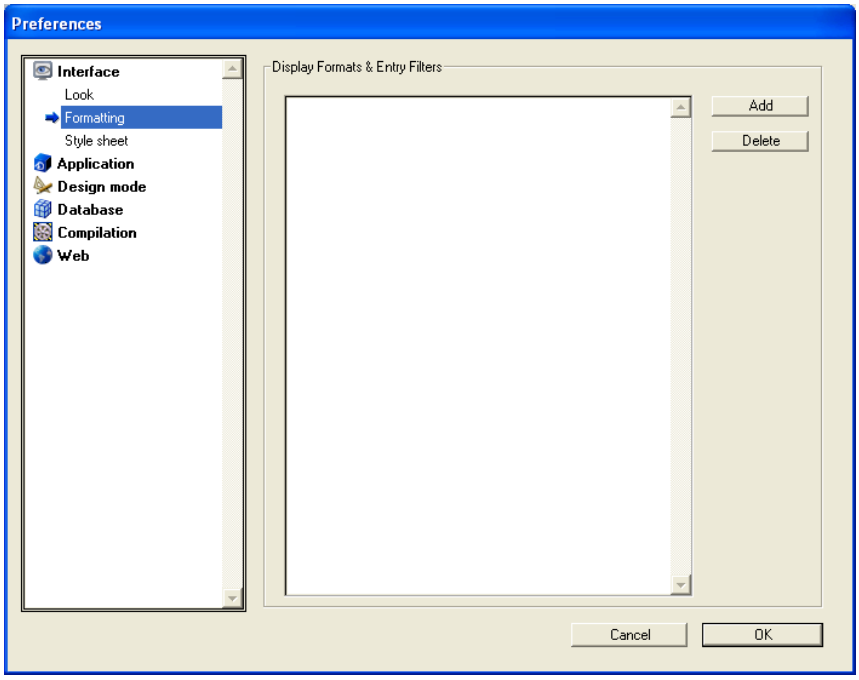
You can monitor the progress of an operation such as indexing records through a numeric display or a graphic thermometer. The thermometer is slower, but easier to read. The numbers are faster, but they do not always refer to the number of records processed. For example, when 4th Dimension performs a sort, the numbers actually show the number of comparisons made.

Toolbar

The **Show Toolbar** option controls whether the toolbar is displayed in the Design, User and Custom Menus environments. Keep in mind that this setting applies to each database opened with your 4th Dimension application.

Formatting Page

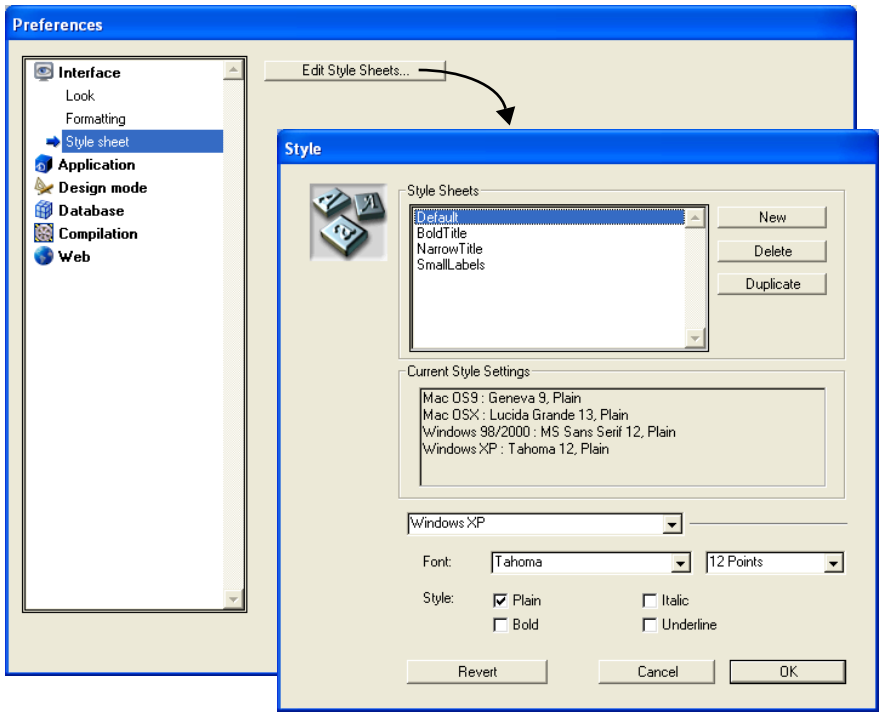
The Formatting page lets you create customized display formats and entry filters:



After you create a custom format or filter, you can refer to it by name instead of having to recreate the format or filter every time. For information on custom formats and filters, see the section “Creating Custom Display Formats and Entry Filters,” page 385.

Style sheet Page

This page lets you access the Style dialog box of your application:



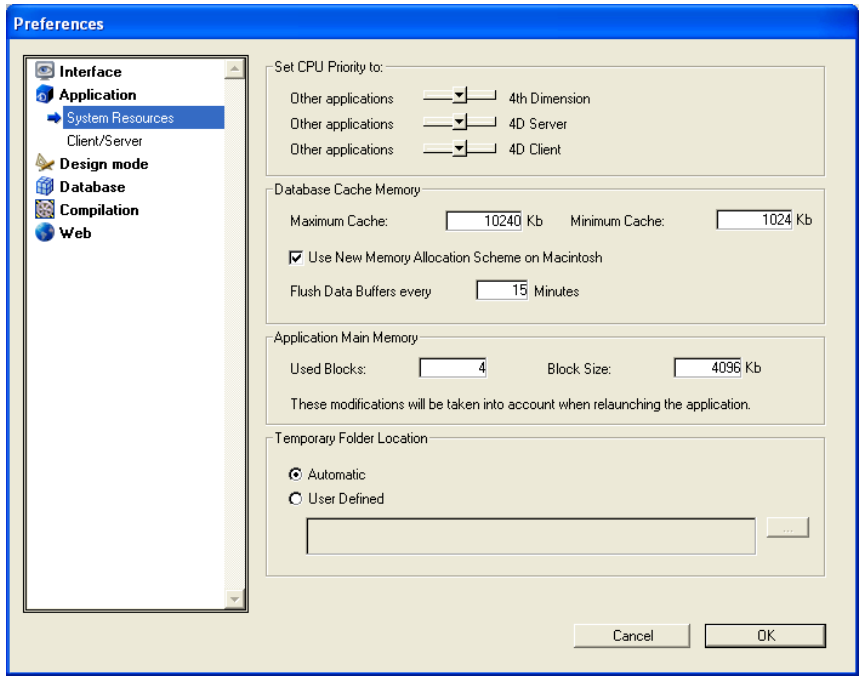
A style sheet groups together a font, font size and style specification. You can use style sheets to specify font attributes when creating or editing forms. When you click the **Edit Style Sheets** button, 4th Dimension displays the Style dialog box. For more information about creating and using styles, see the section “Creating a Style Sheet,” page 260.

Application Theme

The pages of this theme let you configure certain aspects of the internal operation of 4th Dimension as well as specific properties of 4D Server, the client-server version of 4th Dimension.

System Resources Page

The System Resources page lets you optimize the performance of 4th Dimension:



Set CPU priority

This area allows you to modify the frequency with which 4th Dimension applications must “switch back” to other applications.

In fact, 4th Dimension “bunks” with the machine’s operating system that hosts it as well as possibly hosting other applications. Since the architecture of systems on which 4th Dimension is run is not designed to multi-task, the system hands off to an application who then switches back to the system and so on. When an application switches back to the system, it indicates the time limit that it wishes to leave other applications.

Depending on your working environment and your specific needs, you can set the priority thermometer allocated to 4th Dimension with respect to other applications operating on the same machine. For a similar database, it is possible to set the priority for 4th Dimension (single-user)/4D Tools, 4D Server and 4D Client separately. These settings are stored in the structure file.

By default (cursor located in the center of the thermometer), the values correspond to "average" settings. The default values result in a machine time consumption equal to about 20%. In some cases this percentage may not be sufficient; if, for example, you attribute a dedicated machine to your 4D Server, it would be a pity to set aside 80% of machine time to a system that will do nothing else... In this case, it is most likely that raising the "4D Server" cursor priority will improve performance. Conversely, in other cases, it will be a good idea to decrease priority in order to force 4th Dimension to leave time for other applications.

Note It is possible to know and set each setting through programming using the Get database parameter and SET DATABASE PARAMETER commands.

Database Cache Memory

This area allows you to set the application's cache memory.

- **Maximum Cache**: maximum number of KB that you want to allocate to the database cache memory.
- **Minimum Cache**: minimum number of KB that you want to allocate to the database cache memory.
- The **Use New Memory Allocation Scheme on Macintosh** option lets you use system memory for the database cache instead of using that allocated to the application. The memory allocated to 4th Dimension will be used if not enough memory is available at the system level.

You must restart your Macintosh for this memory allocation scheme to take effect.

- **Flush Data Buffers Every … Minutes**: specifies the time period between each automatic saving of the data cache, i.e., its writing to disk. 4th Dimension saves your data at regular intervals. You can specify any time interval between 1 and 500 minutes. As a default, 4th Dimension saves your data every 15 minutes. The application also saves your data each time you change to another environment or exit the application. When you anticipate heavy data entry, consider setting a short time period between saves. In case of a power failure, you will only lose the data entered since the previous save.

If each save involves a noticeable slowing down of disk activity, it is a sign that you should adjust the setting. This slowness means that 4th Dimension is saving many records to disk. A shorter period between saves would therefore be more efficient since each save would involve fewer records and thus be faster.

Application Main Memory

This area allows you to define how much memory to allocate to 4th Dimension for your database while running under Windows. If you modify the parameters, you must restart the application in order for the changes to be taken into account.

The main memory manages the memory allocation of: process stacks, all the Structure elements (forms, methods, lists, etc.), variables, current selections, named selections, sets, plug-ins and transactions. The amount of the Windows main memory equals the block size multiplied by the number of blocks.

The memory blocks are dynamically allocated to the Windows main memory in relation to 4D's requirements.
4th Dimension allocates blocks of memory gradually depending on its requirements. Also, you can set a number of memory blocks that will only be used in case of high activity. This dynamic mechanism lets you avoid using virtual memory which would deteriorate performance drastically. It allows you to use other Windows applications simultaneously, leaving free memory for the system.

- **Used Blocks**: This parameter allows you to set the maximum number of memory blocks that 4D is authorized to load when necessary. This parameter will not allow any value less than 2.
- **Block Size**: This parameter allows you to set the size of a memory block. The size of a block is set by default to 4096 KB.

Note These parameters will apply to all databases opened by the current 4th Dimension application. The 4D Customizer Plus utility lets you set these parameters for all the 4th Dimension applications (4th Dimension, 4D Server, 4D Client, 4D Tools) launched on the machine.

Temporary Folder Location

This area allows you to select the location on your disk where you want 4th Dimension to store the temporary files that are created while the database is running. 4th Dimension mainly uses temporary files for transactions and named selections.

By default, temporary files are managed automatically (**Automatic** option):

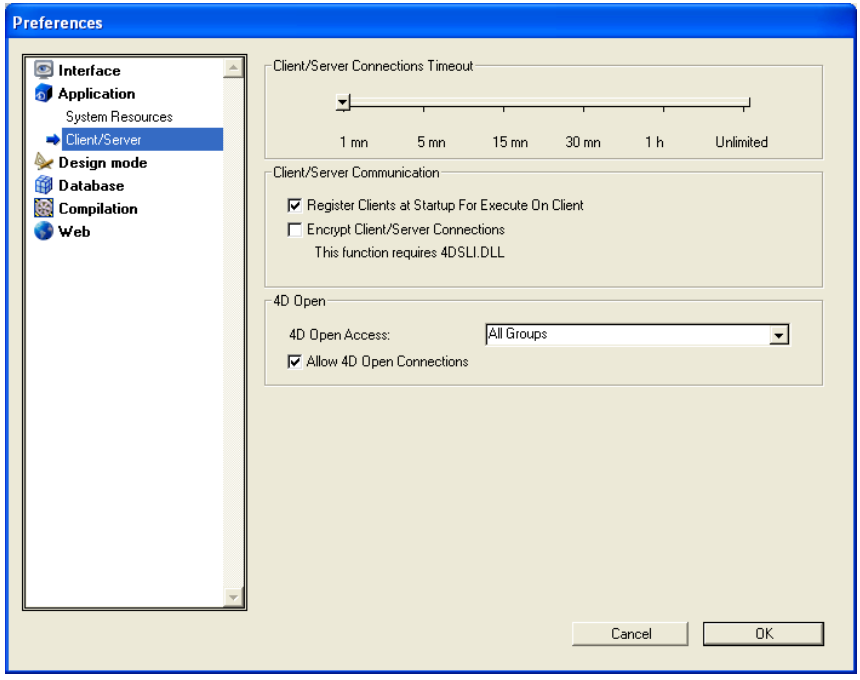
- Under Windows, 4th Dimension places temporary files on drive C,
- Under MacOS, 4th Dimension places temporary files on the local drive with the most available free space (without taking into account any remote drives that may be mounted).

To change the location of the temporary folder, check the **User Defined** option then click on the **[...]** button. A standard Open file dialog box appears, allowing you to select the location where you want to place the temporary folder. This pathname is stored in the 4th Dimension Preferences file.

Note If the pathname is incorrect, the **Automatic** mode is reactivated (without the option being modified in the Preferences dialog box). It is the database developer's responsibility to make sure that the pathname (names of the volumes or folders) is not modified.

Client/Server Page

This page lets you set the 4D Server connection parameters. Naturally, these parameters are only taken into account during client/server operation of the database:



Client/Server Connections Timeout

This thermometer is used to set the *timeout* (period of inactivity beyond which the connection is closed) between 4D Server and the client machines connecting to it.
The **Unlimited** option removes the *timeout*. When this option is selected, client activity control is eliminated and an inactive client will remain connected indefinitely.
When a timeout is selected, the server will close the connection of a client if it does not receive any requests from the latter during the specified time limit.

Client/Server Communication

This area lets you set options relating to communications between the client machines and the server.

- **Register Clients at Startup For Execute On Client:** This option lets 4D Server automatically register each 4D Client machine as soon as it connects to the 4D Server database. Once a client is registered, it can perform any processing that was requested by the server or another client.
  This option is mainly designed for the use of the **Execute on Client** function when run from the Execute Method dialog box in the User environment (for more information about the Execute Method dialog box, refer to the *4th Dimension User Reference Manual*).
  If you want to implement a sophisticated task allocation system, you should consider using the language commands that were designed for that purpose (for more information, refer to the *4th Dimension Language Reference Manual*).
  You can set this option in the Preferences of the client or server machine. In both cases it will apply to each client that connects to the database since it is stored in the structure file of the database. If this option is modified, you have to quit and reconnect all the clients that were already connected in order for the modification to be taken into account.

- **Encrypt Client/Server Connections**: This allows you to activate or disable 4D Server connection encryptions. In fact, the "classic" client/server architecture can benefit from the encryption features proposed by the SSL protocol. Its use allows you to reinforce communication security but it will slow down connections. This option does not require any additional settings.
  By default, this option is not selected.
  For more information, please refer to the *4th Dimension Language Reference* manual.

4D Open

4D Open is the API (Application Programming Interface) that allows non-4D Client applications to connect to 4D Server.

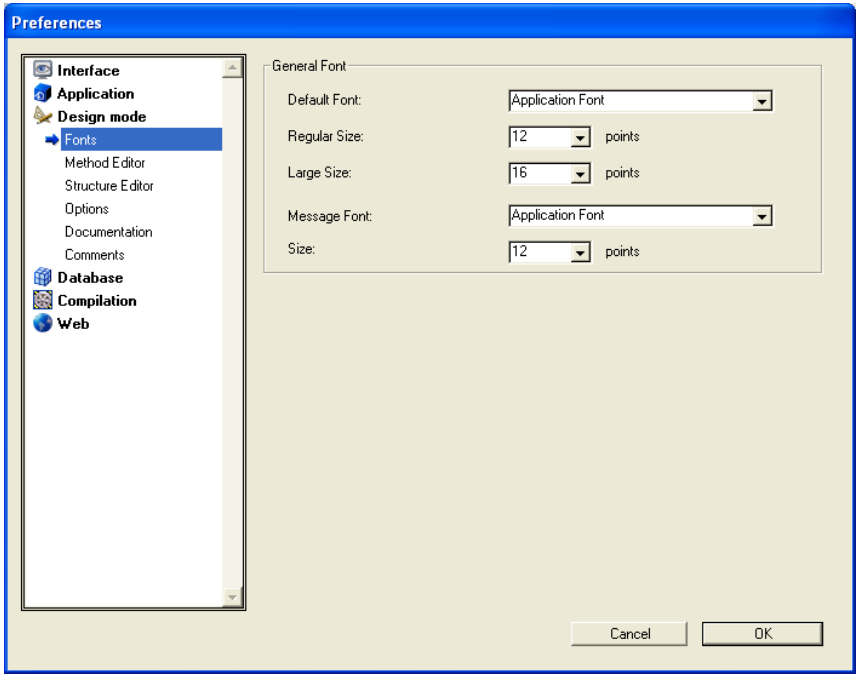
- **Allow 4D Open Connections**: When it is checked, this option gives the group specified by the "4D Open Access" menu the possibility of connecting to 4D Server from a 4D Open application.
- **4D Open Access**: This menu lets you specify the group of users allowed to connect to the 4D Server database via 4D Open, when the "Allow 4D Open Connections" option is checked. For more information about creating groups of users, refer to the chapter "Managing Password Access," page 613.

Design Mode Theme

The pages of this theme let you configure various preferences relating to the interface and to the operation of the windows and editors of the 4th Dimension Design environment.

Fonts Page

This page lets you set the character fonts to be used for the Structure editor, the empty windows of the User environment and the messages:

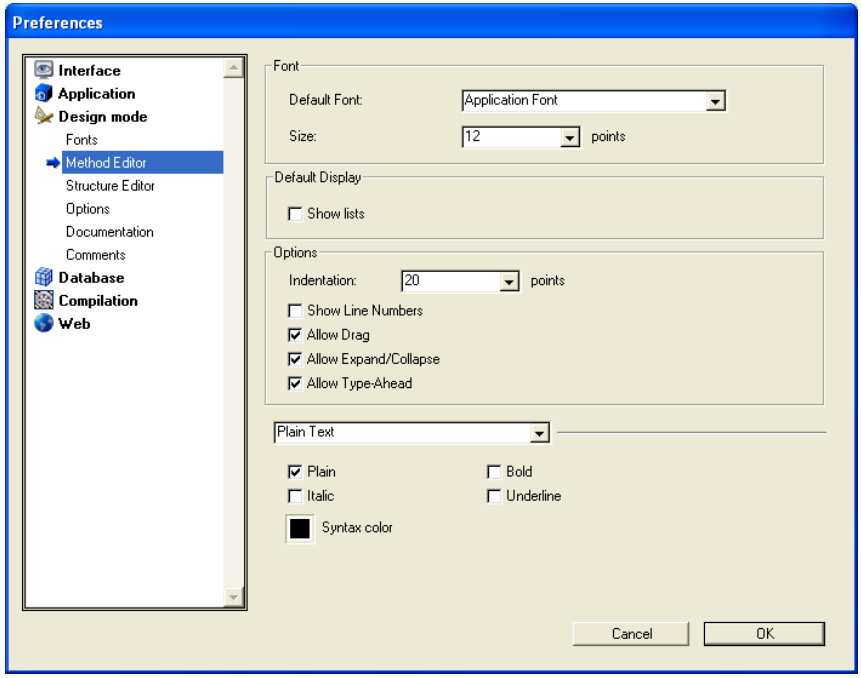


General Font

- **Default Font:** This menu lets you set the character font used in the Structure editor (table and field names) as well as in the empty windows of the User environment (for example, displaying the message "There are no selected records for: [TableName]"). You can set the size of the font for the two different types of locations separately:
  - **Regular Size**: default font size used in the Structure editor.
  - **Large Size**: default font size used in the empty windows of the User environment.
- **Message Font** and **Size**: This option lets you set the font and font size used for messages (displayed more particularly using the MESSAGE command).

Method Editor Page

This page lets you set the interface, default display and operating options of the Method editor:



For more information about the Method editor, refer to the paragraph "Using the Method Editor," page 548.

Font

- **Default Font** and **Size**: These menus let you set the font and font size to be used in the entry area of the Method editor.

Note You can set the font style for each different type of object. For more information, refer to the paragraph "Style and color of syntax elements," page 124.

Default Display

- **Show Lists**: This option lets you choose whether or not to show the lists of objects (Commands, Tables and fields, etc.) by default when the Method editor window is opened. By default, this option is checked and the lists are displayed.

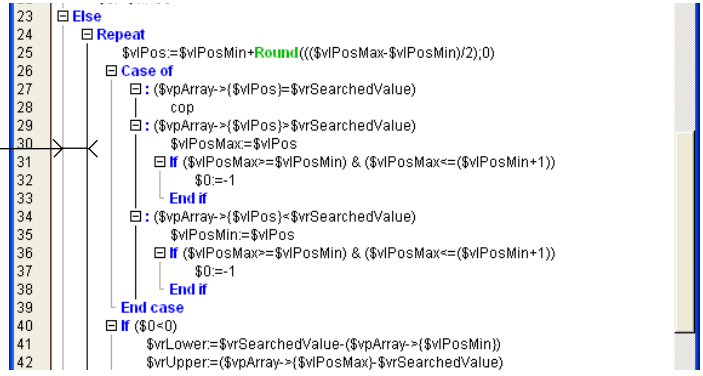
Note It is possible to show or hide the lists for the current window directly from the Method editor. Refer to the paragraph "Using and configuring the Method editor window," page 549.

Options

- **Indentation**: This option lets you change the indentation value for the code in the Method editor. The width must be specified in points (20 by default).

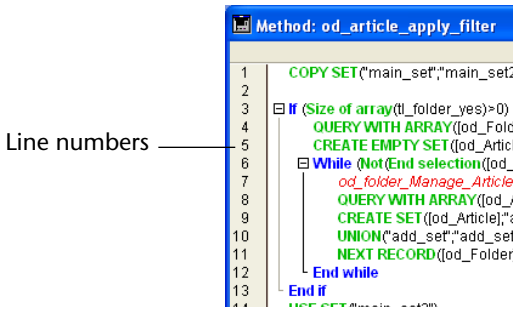
  4th Dimension code is automatically indented in order to reveal its structure:

Default indentation



Modifying this default value can be useful if your methods contain complex algorithms with many levels of embedding. Narrower indentation can be used in order to limit horizontal scrolling.

- **Show Line Numbers**: This option lets you display the line numbers in each window of the Method editor:

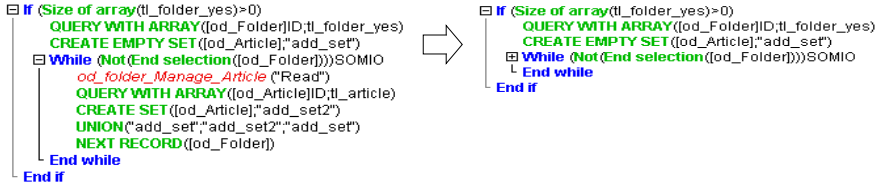


Note It is possible to show/hide the line numbers for the current window directly from the Method editor. Refer to the paragraph “Display of line numbers,” page 556.

- **Allow Drag**: This option (checked by default) lets you enable/disable the drag-and-drop mechanism inside a method or between two methods.

Note This option does not affect the drag-and-drop mechanism of the 4th Dimension Explorer.

- **Allow Expand/Collapse**: This option (checked by default) lets you enable/disable the display of 4D code located inside loops and conditions in the form of hierarchical lists that can be expanded and collapsed:



When this option is not checked, the expand/collapse icons disappear and the code is displayed in expanded form.

- **Allow Type-Ahead**: This option (checked by default) lets you enable/disable the type-ahead function provided in the Method editor of 4th Dimension. For more information about these functions, refer to the paragraph “Type-ahead function,” page 566.

Style and color of syntax elements

The options of this area let you assign a specific style and color to each type of element of the 4D language (fields, tables, variables, parameters, etc.). The combination of different colors and styles for the method elements can prove to be extremely useful for code maintenance.

To define the graphic attributes of a syntax element, select it in the pop-up menu then choose the desired style options. You can apply several different attributes to the same element.
To set a color, click on the selection area in order to make the color palette appear.
You can set the attributes for several elements one after the other — it is not necessary to click **OK** after each modification.

Notes

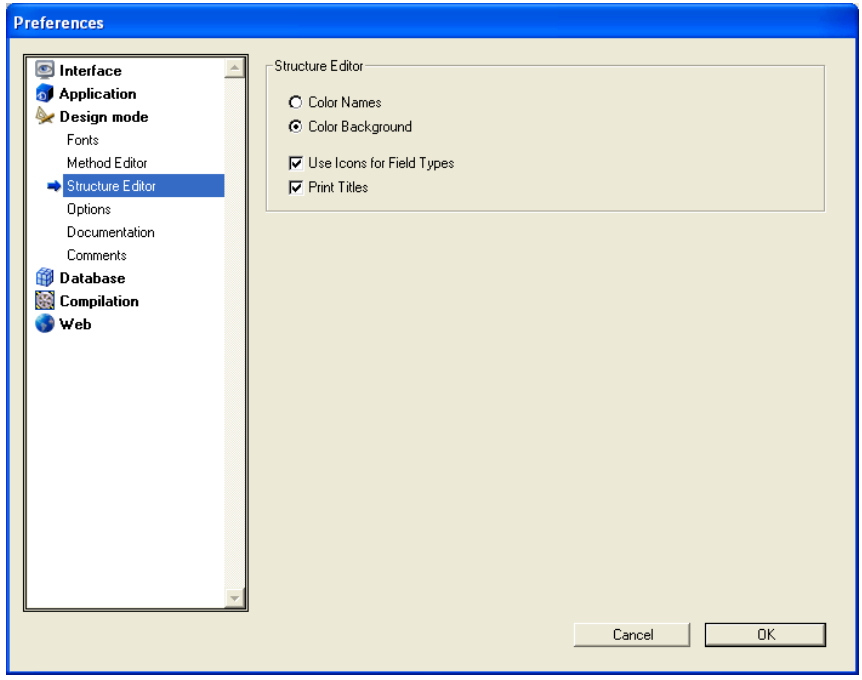
• The "Plain text" element type indicates all the texts not belonging to any other defined type (i.e., symbols, punctuation, literal constants, etc.).
• The "Keywords" element type indicates the programming structures (If/End if, Case of/End case, etc.) that are accessible via Macros.

The available styles and their default use are the following:

- **Normal**
- **Bold** (associated by default with 4D commands, keywords and plug-in commands).
- **Italic** (associated by default with plug-in commands and methods)
- **Underline** (associated by default with predefined constants).

Structure Editor Page

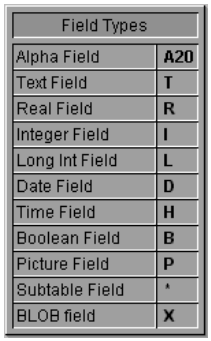
This page lets you set options relating to the display and printing of the 4th Dimension Structure editor window:



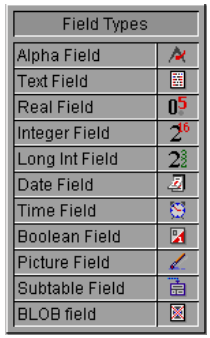
- **Color Names** or **Color Background**: These options let you choose between coloring either the table and field names or the table image in the Structure editor. For information on setting the color, see the sections "Setting the Color of the Table Image," page 171 and "Setting the Color of the Field," page 189.
- **Use Icons for Field Types:** Select this check box to display icons rather than letters to designate the field type in the second column of the table image.

Keep in mind that additional information is provided when the field types are displayed in letter form, i.e., the number of characters in Alpha fields.

Letters



Icons

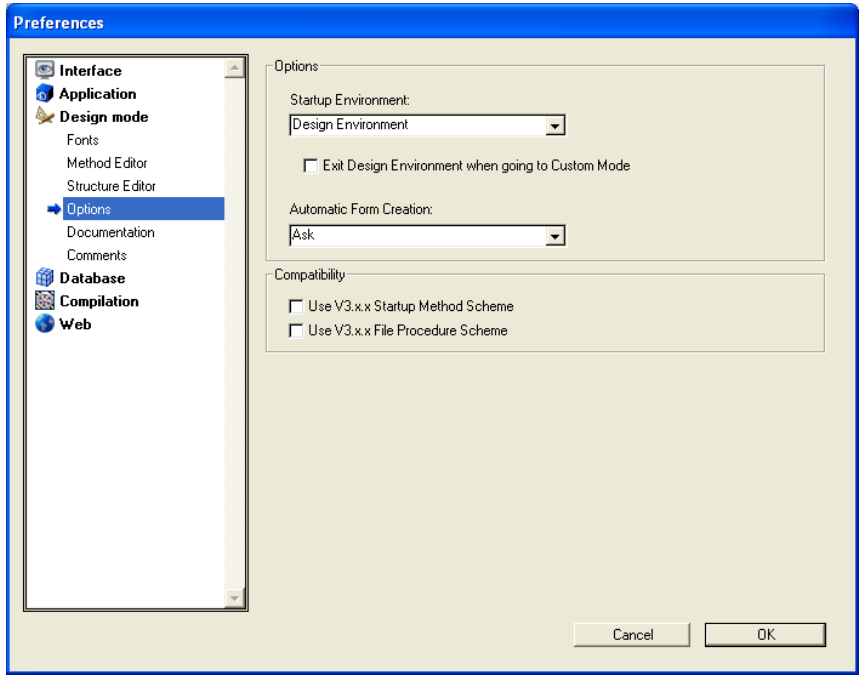


For information on field types, refer to the section “Field Types,” page 176.

- **Print Titles:** Select whether or not to print window titles when you print from the Structure, Form, Method, and Password Access editors. Unless you specify otherwise, 4th Dimension automatically prints window titles, the date on which the window was printed, and the page number.
  If you deselect this option, 4th Dimension omits the title when printing from these editors.

Options Page

This page lets you set various options relating to application startup and to the maintaining of compatibility with previous versions of 4th Dimension:



Options

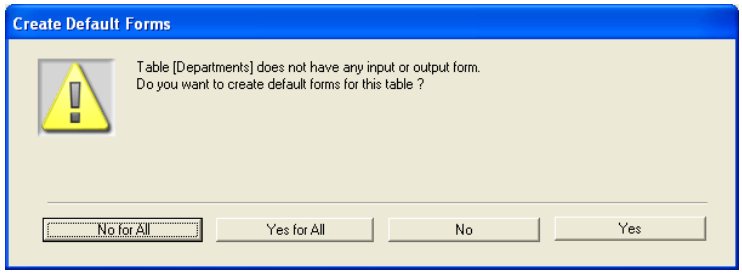
- **Startup Environment:** Select the environment in which you want the database to open. Unless you specify otherwise, by default 4th Dimension opens the database in the Design environment unless the database has been compiled or your user password prevents you from accessing the database in the Design environment. For information on the password access system, see the chapter "Managing Password Access," page 613.

Note The choice between opening a database in compiled or interpreted mode is made in the database opening dialog box (see the paragraph "Opening options," page 31).

- **Exit Design Environment when going to Custom Mode**: If this option is checked, when the user changes to the Custom Menus environment, all the windows of the Design environment are closed. If this option is not checked (default value), the windows of the Design environment will remain visible in the background of the Custom Menus environment.

Note Regardless of whether this option is checked, it remains possible to close the windows of the Design environment by holding down the **Shift** key while selecting either the **User** or **Custom Menus** command in the **Use** menu (see the paragraph "The Design Environment," page 37).

- **Automatic Form Creation**: This menu lets you set the behavior of 4th Dimension when you create a table in the Design environment and then change to the User or Custom Menus environment. By default, 4th Dimension tells you that no form has been created for the new table and then gives you the option of creating default input and output forms automatically:



There are three options available in the menu:

- **Never**: The alert dialog box doesn't appear and no default form is created.
- **Ask**: The alert dialog box appears when no form for the table has been created (default option).
- **Always Yes for All**: The alert dialog box doesn't appear, but default forms are created for all the tables automatically.

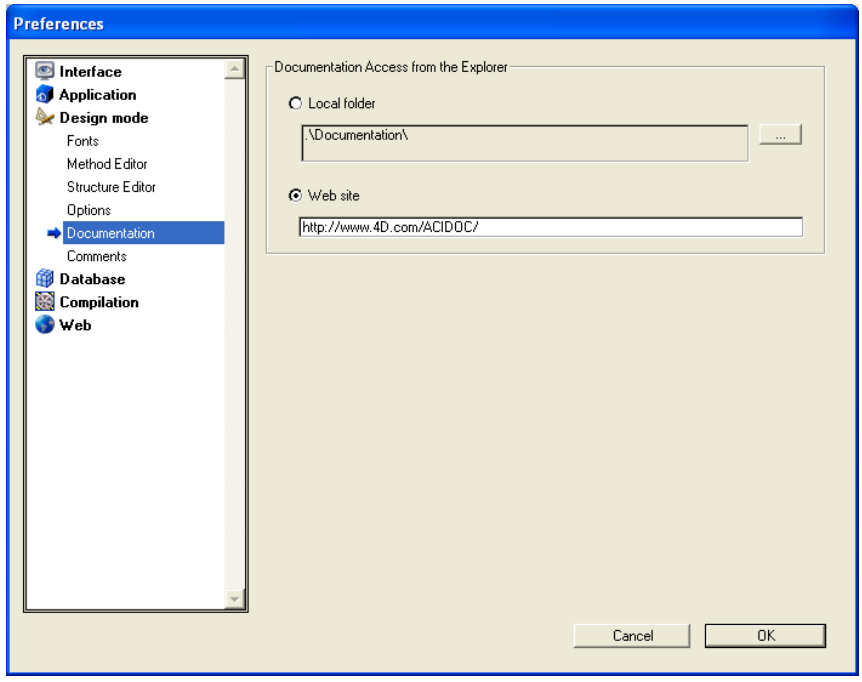
Compatibility

- **Use V3.x.x Startup Method Scheme:** This option allows you to let the *STARTUP* method be called automatically when the database is opened, as in 4th Dimension prior to version 6.0. The database methods are only active if this property is *not* selected. If you are converting a database created with a version prior to version 6 and want to use the new Database methods architecture, you can copy your *STARTUP* procedure into the On Startup database method and deselect this property. For more information about database methods, see the section "Database Methods," page 521.

- **Use V3.x.x File Procedure Scheme:** If this property is selected, Table methods (triggers) run according to the rules established for File procedures in earlier releases of 4th Dimension. File procedures were only executed for input forms and were executed each time a form element was used (e.g., when a button was pressed or data was entered in a field). This property can be used for either converted databases or databases created with the current release of 4th Dimension.

Documentation Page

This page lets you set the access to on-line documentation from the Explorer of 4th Dimension:



Note For more information about access to the on-line documentation from the Explorer, refer to the paragraph "Commands Page," page 77.

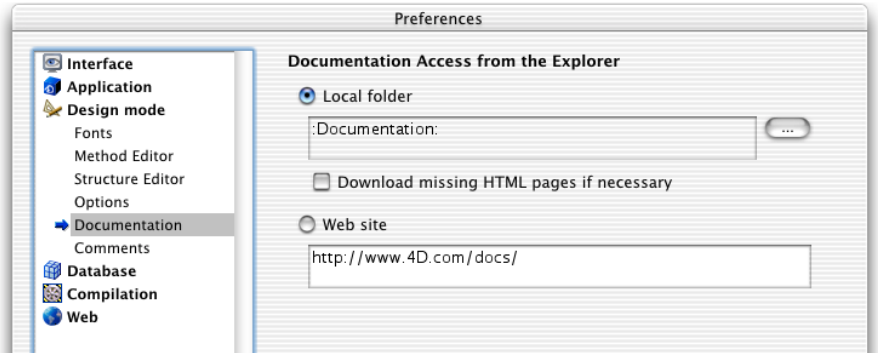
- **Local folder/ Web site**: This option is used to indicate the location from which HTML documentation pages are loaded:
  - when the user double-clicks on a command in the Explorer (Windows and MacOS);
  - when a command is selected in the Explorer (MacOS only).

If you select the **Local folder** option, 4th Dimension will look for the HTML pages in the specified folder. By default, this location is the Documentation folder located next to the 4D application (under MacOS, next to the 4D software package).
The indicated pathname is relative to the 4D application. You can modify this location as desired. The HTML documentation can be located on another volume, a CD-Rom, etc. To designate another location, click on the **[...]** button located next to the entry area and choose a documentation root folder (folder containing the 4DDOCFR.HTM, 4DDOCUS.HTM and 4DDOCGM.HTM files).

If you select the **Web site** option, 4th Dimension will look for the HTML pages at the specified URL. The default URL corresponds to the standard 4D 2003 documentation URL on the 4D, Inc. Internet site. This URL can be modified as desired.

- **Download missing HTML pages if necessary** (MacOS only)

MacOS



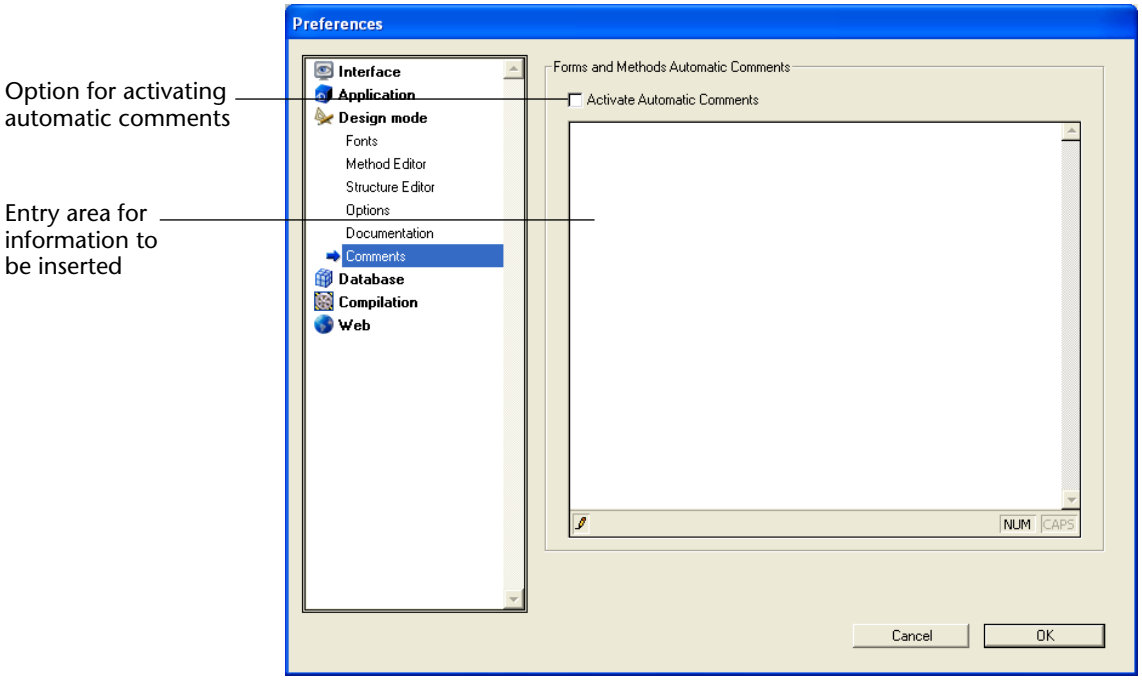
This option is used to activate (or deactivate) automatic downloading of the HTML documentation pages from the Internet when they are not found in the “Local folder.” This enables you to automatically integrate new 4D commands into your documentation or to fill the local folder gradually according to your needs.
By default, this option is not checked: when you have chosen to load the documentation from the **Local folder**, if the page corresponding to a 4D command is not found in this folder, 4th Dimension displays an error page.

However, if the **Download missing HTML pages if necessary** option is checked, 4th Dimension will download the page concerning the command from the URL specified in the "Web Site" area. Once it is downloaded, the page is stored in the local folder.

4D Server The settings for access to on-line documentation are specific to each client machine.

Comments Page

The Comments page lets you activate and set the automatic comments system for your database:



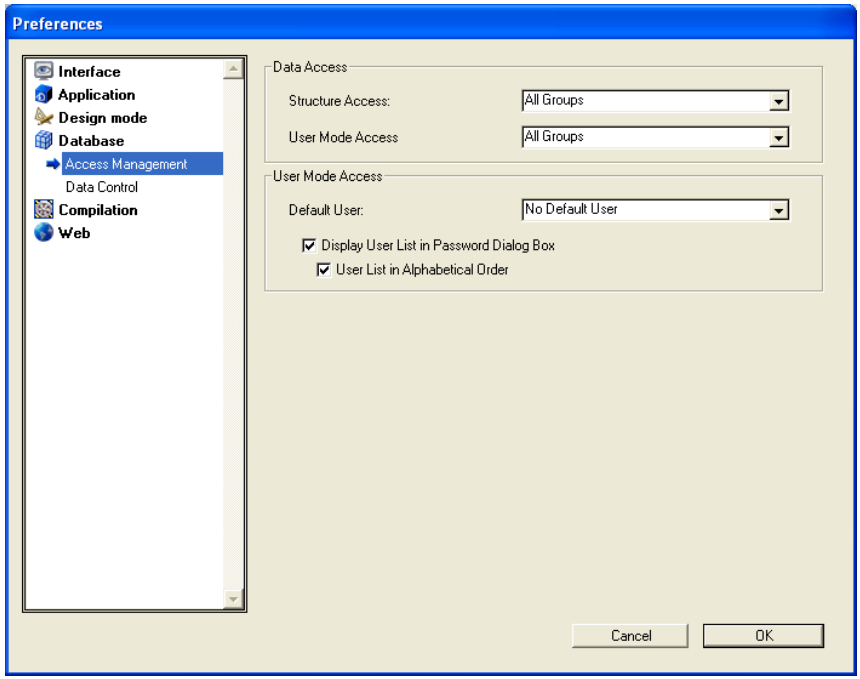
When the **Activate Automatic Comments** option is selected, comments that are entered in the entry area will appear for each method or form that is either created or modified in the database. The comments can be displayed in the Explorer. For more information, refer to the paragraph "Inserting Automatic Comments," page 88.

Database Theme

The pages of this theme let you protect access to the database and set various options relating to the processing and integrity of the data.

Access Management Page

This page lets you control access to the Structure and User environments of 4th Dimension and to set the User Mode Access conditions:



Data Access

- **Structure Access**: Gives the specified group the ability to enter the Design environment. For information about users and groups, see the chapter “Managing Password Access,” page 613.

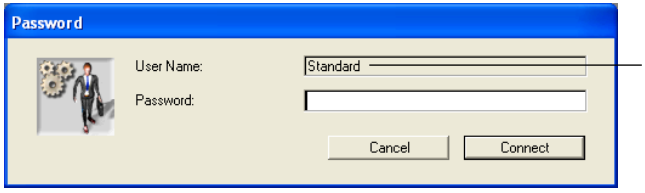
Note Setting an access group in the Design environment lets you deactivate the **Create table** option in the data import dialog box. For more information about this dialog box, refer to the *User Reference* manual.

- **User Mode Access**: Gives the specified group the ability to access the User environment. Any user who does not belong to this group will not be able to access the User environment from the Customs Menu environment, nor using the menu command or the standard key combinations (**Option+F** under MacOS, **Alt+F4** or clicking on the close box under Windows). Any attempt from an unauthorized user to switch to the User environment will result in the 4th Dimension application quitting.
  - The Designer and Administrator always have access to the User environment even if they are not explicitly part of the group that has access to the User environment.
  - A user that has access to the Design environment always has access to the User environment, even if they are not explicitly part of the group that has access to the User environment.

  For more information about user and user groups, refer to the chapter "Managing Password Access," page 613.

User Mode Access

- **Default User**: When a Default User has been defined, every user that opens the database or logs onto it has the same access privileges and restrictions defined for this Default User. A default user is not required to enter a user name. Moreover, if you have not associated a password with the Default User, the Password dialog box no longer appears and the database opens directly.
  This option simplifies access to the database while maintaining a complete data control system.
  - If you have associated a password with the Default User, a dialog box appears when the database is opened. The Default User's name appears in the User Name area and the Default User's password must be entered:



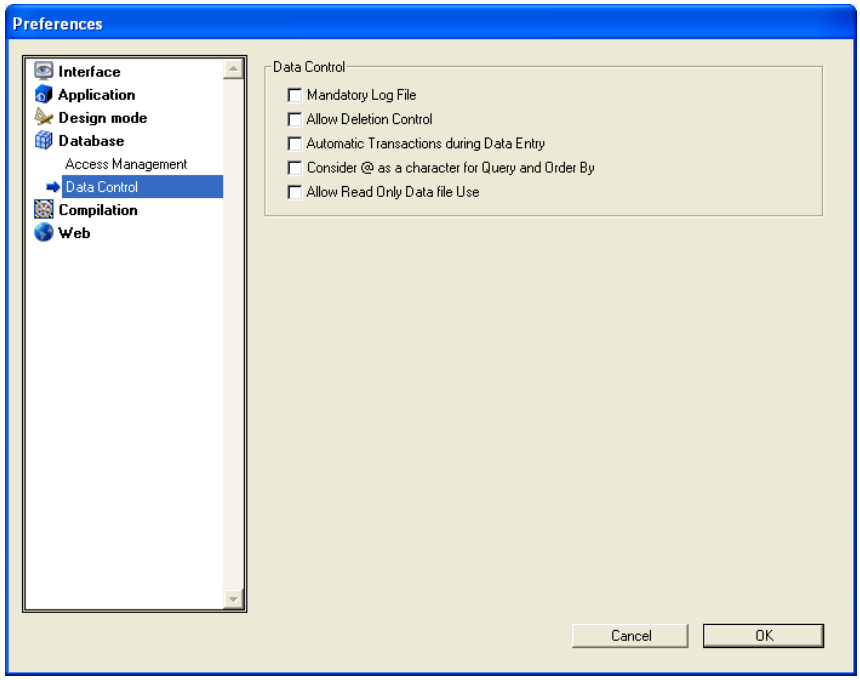
  - If you haven't associated a password with the Default User, the above dialog box will not appear.

Note You can "force" the display of the password entry dialog box when the "Default User" mode is active, for instance in order to connect as Administrator or Designer. To do so, press the **Shift** key while opening the database or connecting to it.

- **Display User List in Password Dialog Box**: If this option is checked, users must choose their name from the list of users and enter their password in the password entry dialog box. If it is not checked, users must enter both their name and password. For more information about the two versions of the password dialog box, see the section "Access System Overview," page 614.
- **User List in Alphabetical Order** (only available if the previous option is checked): When this option is checked, the list of users in the password entry dialog box is sorted by alphabetical order.

Data Control Page

This page lets you set various options relating to data integrity:



- **Mandatory Log File:** This option requires that the database be opened with a log file. The log file is designed to keep track of changes to a database since the last backup. The 4D Backup plug-in is required to use this feature.

- **Allow Deletion Control:** This option allows you to use the Deletion control options in the Relation Properties window for each relation you define. If the **Allow Deletion Control** check box is not selected, the Deletion control options are disabled. For more information, see the section "Relation Types," page 214.

- **Automatic Transactions during Data Entry:** Automatically start a multi-transaction when an input form is first opened when the form has a subform. *This option is intended only for 4D First users who upgrade to 4th Dimension. It should not be used for 4th Dimension custom applications (i.e., databases that work in the Custom Menus environment), where it is preferable that the transactions be handled by the developer.*

- **Consider @ as a character for Query and Order By:** This option allows you to set how the *at* sign "@" will be interpreted when used in a query or a sort. This setting applies only when the *at* sign is located in a word.
  When this option is not checked (default value), the *at* sign is used as the wildcard character. For example, 'F@w" finds words that begin with 'F', contain any number of letters, and end with a 'w'. (see the *User Reference* manual).
  When the option is checked, the *at* sign is regarded as a single character—the *at* sign "@". This setting is especially useful when searching for E-mail addresses, where the @ sign is used internally.
  This option has an influence on searches, sorts, string comparisons, as well as on data stored in tables and data found in memory, like arrays. Fields and variables of the alpha (indexed or not) and text type are concerned with how the @ character is interpreted in searches and sorts.

Notes

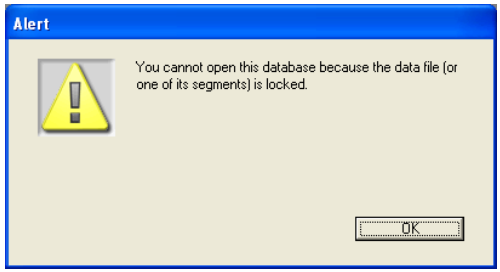
• For searches, it is important to note that if the search criteria **begins** or **ends** with @, the "@" character will be treated as a wildcard. Only if the "@" character is placed **in the middle of** a word (for example: bill@cgi.com) will 4th Dimension treat it differently.

• This option can also have an influence on the behavior of the commands in the "Object Properties" theme that accept the wildcard character ("@") in the object parameter. Please refer to the *4th Dimension Language Reference* manual.

• For security reasons, only the Administrator or Designer of the database can modify this parameter.

If you modify this setting, you have to quit and reopen the database to make the change effective. Once the database is reopened, all of the database's indexes are automatically re-indexed.

- **Allow Read Only Data file Use**: This option allows configuration of the application operation when opening a locked data file at the operating system level.
  4th Dimension includes a mechanism that automatically prevents the opening of a database when its data file, or one of its segments, is locked. In this case, when this detection option is activated, 4D displays a warning message and does not open the database:



  Unless this option is checked, it is not possible to open a database when its data file is locked (default operation for 4D databases created with version 2003 or later). Note that this option is only applied to the open database and not to all the databases opened with the 4D application.

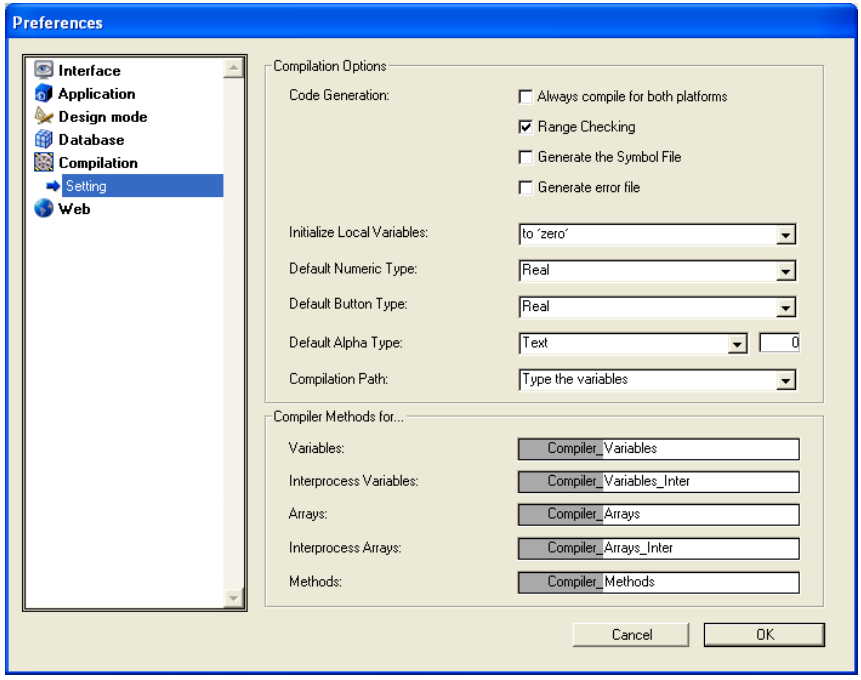
  - **About locked files**
    Locked files may be read but their content cannot be modified. For example, files are locked when they are stored on a non-rewritable support (CD-Rom type) or when they are recopied from this type of support. 4th Dimension can work in a transparent manner with locked data files, which allows, more particularly, the use of databases stored on CD-Rom.
    However, this operation runs the risk of inadvertent use of a locked data file in which modifications will not be saved. This is why by default 4th Dimension does not allow databases with a locked data file to be opened.

Compilation Theme

This theme consists of a single page that lets you set the parameters to be used for database compilation. For more information about database compilation, refer to the chapter “Compiling a Database,” page 689.
These settings will be applied to all the databases that are opened with the current 4th Dimension application.

Setting Page

This page contains the parameters relating to database compilation:



Compilation options

This area groups the generic options used during the compilation process.

- **Always compile for both platforms**: By default, when this option is not checked, 4th Dimension generates compiled code corresponding to the platform on which the application is running.
If you check this option, 4th Dimension will generate both Pentium (Windows) and PowerPC (MacOS) compiled code, whatever the running platform.

- **Range Checking**: used to activate/deactivate range checking. Range checking is an additional verification that checks the code in situ and according to the state of database objects at a given moment.

  For more information about range checking, refer to the paragraph "Range checking," page 705.

- **Generate the Symbol File**: This option is used to generate an ASCII type file (text only) containing the list of variables along with their type and the method from which their type has been inferred. The symbol file also contains the list of your methods and functions along with the type of their parameters and the type of result, if any.
  The file is placed in the folder containing the database structure and is named:
  - under Windows, DatabaseName.sym,
  - under MacOS, DatabaseName.symb.

  For more information about the symbol file, refer to the paragraph "Symbol file," page 701.

- **Generate error file**: This option is used to generate the error file at the time of syntax checking. It lists general errors as well as errors linked to a specific line, and warnings.
  Any errors detected by the compiler are automatically accessible in the **Method** menu of 4th Dimension 2003. However, having an error file that can be transmitted from one machine to another can be useful, particularly in a situation where several different developers are working together in a client/server environment.
  The error file is generated in XML format in order to facilitate automatic parsing of its contents. It also allows the creation of customized error display interfaces.
  The error file is automatically named DatabaseName.xml and is created:
  - with 4th Dimension, next to the structure file of the database,
  - with 4D Client, next to the .exe file of the 4D Client application (Windows) or next to the 4D Client software package (MacOS).

  For more information about the error file, refer to the paragraph "Error file," page 703.

- **Initialize Local Variables**: This option is used to define the local variable initialization mode at the beginning of methods:
  - **to 'zero'**: Variables are reset to zero by default (empty string for character strings, 0 for numbers...).
  - **to a random value**: The compiler assigns a random value, always the same, (1919382119 for longints, "rgrg" for character strings, True for Booleans...) to variables. This option enables you to pinpoint local variables that you have forgotten to initialize.
  - **no**: The compiler does not initialize the variables. In this way, you gain time during database execution, provided that your initialization was correct.
- **Default Numeric Type**: This option is used to force numeric typing in an unambiguous manner, either in real or longint. It has no priority over any directives that may have been placed in your database. You can optimize the running of your database by choosing the Longint type.
- **Default Button Type**: This option is used to force button typing in an unambiguous manner, either in real or longint. It has no priority over any directives that may have been placed in your database. It concerns standard buttons as well as the following objects: check boxes, 3D check boxes, highlight buttons, invisible buttons, 3D buttons, picture buttons, button grids, radio buttons, 3D radio buttons, picture radio buttons, picture pop-up menus, hierarchical pop-up menus and pop-up/drop-down lists.
- **Default Alpha Type**: This option is used to force character string typing in an unambiguous manner, either in text or fixed string. It has no priority over any directives that may have been placed in your database. If you choose to give character strings the **Fixed string** type by default, the enterable area allows you to indicate the length of these strings to the compiler (you must enter a value between 2 and 80). Choosing the Fixed string type lets you optimize the running of your databases.
- **Compilation Path**: This option is used to define the number of passes performed by the compiler and thus the duration of compilation.
  - **Type the variables**: passes by all the stages that make compilation possible.

- **Process and interprocess are typed**: The pass for typing process and interprocess variables is not carried out. This option can be used when you have already carried out the typing of all your process and interprocess variables either yourself or using the function for the automatic generation of compiler methods.
- **All variables are typed**: The pass for typing local, process and interprocess variables is not carried out. Use this option when you are certain that all the process, interprocess and local variables have been clearly typed.

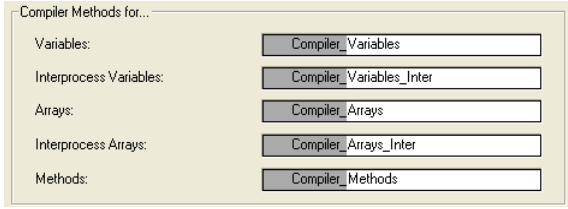
Compiler Methods

This area lets you rename the Compiler methods that are generated automatically by the compiler. These methods group together all the variable typing declarations, process and interprocess arrays, as well as the local variable declaration methods. These methods are generated using the compilation window. For more information, refer to the paragraph "Compiler window," page 694.

Up to 5 compiler methods may be generated; a compiler method is only generated if the database contains the corresponding items:

- **Variables**: groups together process variable declarations;
- **Interprocess Variables**: groups together interprocess variable declarations;
- **Arrays**: groups together process array declarations;
- **Interprocess Arrays**: groups together interprocess array declarations;
- **Methods**: groups together local variable declarations designating method parameters (for instance, C_INTEGER(mymethod;$1)).

You can rename each of these methods in the corresponding areas:



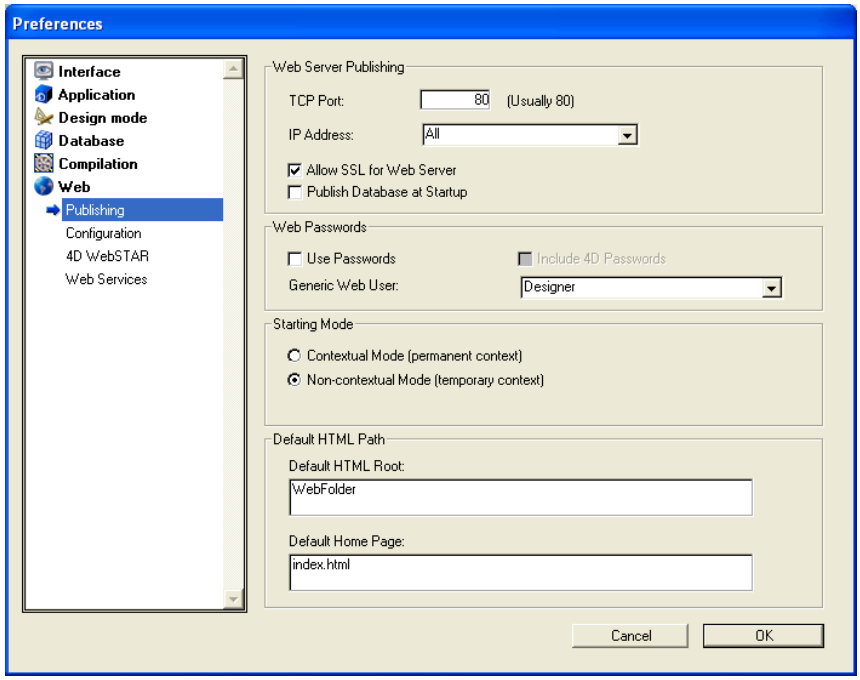
Nevertheless, they will always be preceded by the label "*Compiler_*" (non-modifiable). The name of each method (prefix included) must be unique and no longer than 31 characters. Extended ASCII code characters (accented characters, typographical symbols, etc.) are not allowed.

Web Theme

The pages of this theme let you set the operation of the 4th Dimension integrated Web server (security, startup, connections, etc.), as well as the use or publishing of Web Services. For more information about the Web server, refer to the "Web Server" chapter of the *4th Dimension Language Reference* manual. For more information about Web Services, refer to the chapter "Publishing or Using Web Services," page 723.

Publishing Page

This page groups together the parameters relating to publishing, security and the default HTML pathnames of the Web server:



Web Server Publishing

- **Port TCP**: Set the TCP port to be used when the Web server is started. The default value is 80. Setting the TCP port allows you to run several Web servers on the same computer. To do so, select different TCP ports for each Web server.

This option also allows you to let the OS provide Web services on port 80, while you are using 4th Dimension as a Web server on another port.

Note When the TCP port is different from the default port, you must add the port number to the IP address in order to be able to access the Web server. The syntax to use is *IPaddress:TCPport*. For example,123.45.67.89:8080 indicates that the TCP port used is 8080.

- **IP Address**: This option allows you to specify the IP address on which the Web server will receive the HTTP requests. By default, the Web server answers on all the IP addresses (**All** option).
  The pop-up menu automatically displays all IP addresses present on the machine. If you want the server to only respond to requests sent to a specific TCP/IP address, simply select this address from the menu. This feature is for 4D Web Servers located on machines with multiple TCP/IP addresses, which is frequently the case for most Internet host providers.
- **Allow SSL for Web Server**: This allows you to activate or disable SSL protocol usage for Web server connections. By default, this option is selected. The TCP port used for SSL connections is port 443.
  You can deselect this option if you choose not to use SSL functions in your Web server or if another Web server authorizing secured connections is active on the same machine. For more information, please refer to the *4th Dimension Language Reference* manual.
- **Publish Database at Startup:** This option chooses whether or not to automatically publish the current database on the Web. If this box is checked, 4th Dimension accepts Web connections. If this box is not checked, the database is not published by default. However, you can publish a database at any time in User environment or using language.

Web Passwords

- **Use Passwords**: This option activates the Web server's password system. For each connection, a dialog box appears on the browser so that the user can enter his name and password. These two values, as well as the connection parameters (IP address and port, URL...) are sent to the On Web Authentication database method so that you can process them. If the On Web Authentication database method does not exist, the connection is rejected.

- **Include 4D Passwords:** This option is enabled only when the **Use Passwords** option is selected. It allows you to use, instead of or in addition to your own password system, the password system of the database (as defined in 4th Dimension).

Notes
• With the 4D Client Web server, every site published by 4D Client machines will share the same table of users. In fact, the validation of users/passwords is carried out by the 4D Server application.
• Passwords entered by the users are not encrypted in the HTTP requests (Basic mode).

The Web access control system consists of these options and the On Web Authentication database method. For more information, refer to the "Connection Security" section of the *4th Dimension Language Reference* manual.

- **Generic Web User**: When a user has been set in this menu, each browser that connects to the database will benefit from the access rights and restrictions associated with this user. This provides you with a simple way of controlling the browser's access to different parts of the database. By default, the Generic Web User is the Designer and browsers have full access to the entire database.

Note Do not confuse this option, which allows you to restrict the browser's access to different parts of the database (tables, menus, etc.), with the Web server's connection control system, managed by the password system and the On Web Authentication database method (see above).

The only exception to this operation is when the **Include 4D Passwords** option is selected and the user who connects is defined in 4th Dimension's Password editor. In that case, the **Generic Web User** option is ignored and the users connect under their own access privileges.

Starting Mode

This option lets you set the mode, **contextual** or **non-contextual**, in which the Web server must start. By default, the Web server starts in non-contextual mode.
For more information, refer to the "Using the Contextual Mode" section of the *4th Dimension Language Reference* manual.

Note This option is not available with 4D Client because the contextual mode is not supported by the 4D Client Web server.

- **Reuse Temporary Contexts**: This option (found only in the Preferences of 4D Client) is used to optimize Web server operation in 4D Client by recycling Web processes created for handling previous Web requests. In fact, the Web server of 4D Client needs a specific Web process for the handling of each Web request; when necessary, this process connects to the 4D Server machine in order to access the data and database engine. It then generates a temporary context using its own variables, selections, etc. Once the request has been dealt with, this process is killed.

  When the **Reuse Temporary Contexts** option is checked, 4D maintains the specific Web processes created on 4D Client and reuses them for subsequent requests. By removing the process creation stage, Web server performance is improved. In return, you must make sure in this case to systematically initialize the variables used in 4D methods in order to avoid getting incorrect results. Similarly, it is necessary to erase any current selections or records defined during the previous request.

Default HTML Path

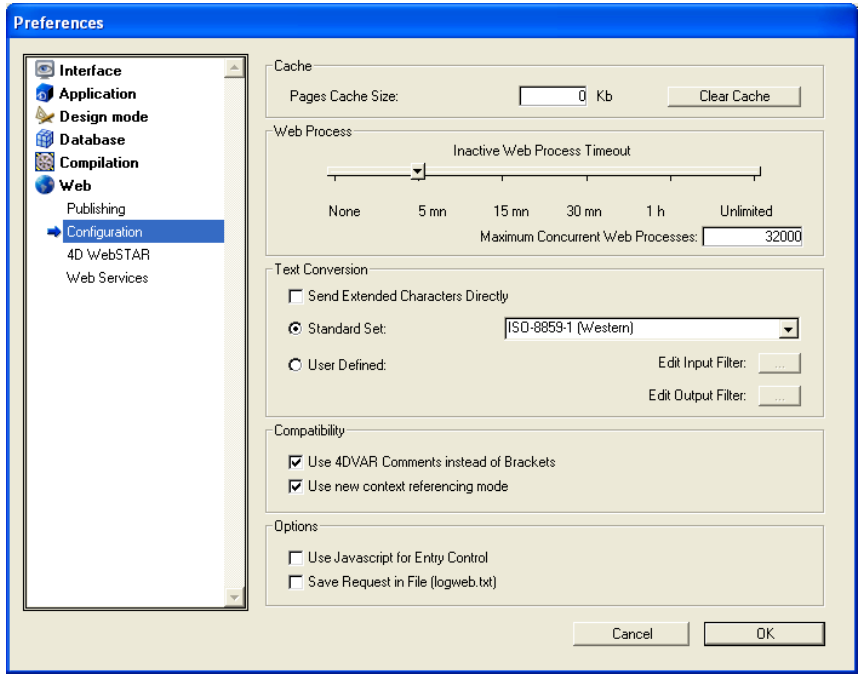
- **Default HTML Root**: This option allows you to define the folder in which 4D will search for the static HTML pages and the pictures to send to the browsers. Moreover, the HTML root folder defines the hierarchical level above which the files will not be accessible.
  By default, 4D defines a default HTML root folder named "WebFolder." This setting automatically activates the access restriction system. If it does not already exist, the folder is created on disk when the Web server is launched for the first time. If you keep the default location, the root folder is created:
  - with 4th Dimension and 4D Server, at the same level as the database structure file.
  - with 4D Client, at the same level as the 4D Client .exe file (under Windows) or the software package (under MacOS).

  To change the location of the Default HTML Root folder, enter the pathname relative to the folder that you want to set as the new location. You must use HTML syntax (Unix).
  For more information, refer to the "Connection Security" section of the *4th Dimension Language Reference* manual.

- **Default Home Page**: This option lets you set a default home page for all browsers that connect to the database.
  For more information, refer to the “Web Server Settings” section of the *4th Dimension Language Reference* manual.

Configuration Page

This page groups together the parameters relating to the internal operation of the 4th Dimension Web server:



Cache

- **Pages Cache Size**: This option lets you activate and set the size of the 4D Web Server cache — allowing you to load static pages, GIF images, JPEG images (<100 KB) and style sheets. Using the cache allows you to significantly increase the Web server’s performance when sending static pages. The cache is shared by all the Web processes.

  By default, the cache of the static pages is not enabled (its size equals 0). The value you set depends on the number and size of your Web site’s static pages, as well as the resources available on the host machine.

Once the cache has been enabled, the 4D Web server first looks in the cache for the page requested by the browser. If it finds the page, it sends it immediately. If not, 4th Dimension loads the page from disk and places it in the cache. When the cache is full and additional space is required, 4th Dimension "unloads" the oldest pages first, among the least requested ones.

- **Clear Cache**: Clicking this button clears the cache of the pages and images that it contains (if, for example, you have modified a static page and you want to reload it in the cache).

Web Process

- **Inactive Web Process Timeout**: lets you set the *timeout* (period of inactivity beyond which the connection will be closed) for Web connection processes (contextual mode only).
  The **Unlimited** option eliminates activity control of the browsers. An inactive client will remain connected indefinitely. However, when a timeout is set, the Web server will close the connection if it does not receive any requests from the client within the specified time interval. For more information about this, refer to the "Using the Contextual Mode" section of the *4th Dimension Language Reference* manual.
- **Maximum Concurrent Web Processes:** defines the limit (strictly superior) of the number of Web processes created by the Web server. This setting includes all types of Web processes: contextual, non-contextual or those belonging to the pool[1] of processes. This setting allows you to avoid 4D Web server saturation that is possible when there is a massive number of requests or an excessive demand of context creations.
  By default, this number is set to 32,000 (in other words, up to 31,999 Web processes can be simultaneously created). You can set any value between 10 and 32,000. In theory, the maximum number of processes is the result of dividing Available memory by Web process pile size.

1. The "pool" of Web processes allows you to increase the Web server reactivity in non-contextual mode. This reserve is resized with a minimum (0 by default) and a maximum (10 by default) of processes to recycle. These values can be modified using the **SET DATABASE PARAMETER** command (selectors 6 and 7). Once the maximum number of pool Web processes has been set, if it is lower than the upper limit of the "pool," this limit will be lowered to the maximum number of Web processes.

Another solution consists of viewing the information on the Web processes displayed in the Runtime Explorer: the current number of Web processes and the maximum number reached since the Web server launch are indicated (see the section “Watch Page,” page 91).
When the maximum number of concurrent Web processes is reached, 4D will not create any other new processes and returns HTTP status 503 - Service Unavailable to each new request.

Note The maximum number of Web processes can also be defined using the **SET DATABASE PARAMETER** command.

Text Conversion

- **Send Extended Characters Directly**: By default, the 4th Dimension Web server converts extended ASCII characters found on the Web pages to HTML standards before sending them. They are then interpreted by the browsers. This option lets you send extended ASCII characters “as is,” without conversion into HTML entities.
This option can save you a considerable amount of time on foreign systems (in particular, Japanese ones).

- **Standard Set:** This option allows you to choose the character set 4th Dimension uses to communicate with Web browsers connecting to the database. The value selected in the drop-down list will determine the conversion of ASCII characters that 4D Web server performs when sending or receiving an HTML document (dynamic or static pages). The default character set for the US version of 4th Dimension is ISO-8859-1, which corresponds to the standard Western (Latin1) encoding. 4th Dimension lets the Web browser know which encoding is to be used for HTML information. It is therefore not necessary to set this particular setting in each browser— except for the **x-user-defined** value.

Note The set of characters defined in this menu is also used for HTML exporting of quick reports and for the Mac to ISO language command.

- The **User Defined** option is equivalent to the selection of **x-user-defined** encoding in the previous menu. This option is designed for countries that use a specific alphabet (Iceland, Greece, and so on). This setting permits the use of any specific encoding.
When this value is selected, the **Edit Input Filter** and **Edit Output Filter** buttons are enabled. These buttons allow you to edit the input and output conversion tables for ASCII characters (Web filters).

Keep in mind that when this option is selected, 4th Dimension does NOT let the browser know about the encoding it is using and it should thus be selected by the user when he connects to the database.

Compatibility

- **Use 4DVAR Comments instead of Brackets**: This option allows you to define the notation to use when inserting 4th Dimension variables on static pages.
  - When the box is checked, the syntax you need to use is the standard HTML notation (<!--4DVAR MYVAR-->)[1].
  - When the option is not checked (default value), the syntax you need to use is the notation with square brackets ([MYVAR]) — which is a proprietary solution used in former versions of the 4th Dimension Web server.
- **Use new context referencing mode**: When this option is selected (default value), the 4D Web server places the context number in the basic URL of the HTML documents being sent.
  With the former system (option not checked), the 4th Dimension Web server sends the context number for each element of a page to the browser, which slows down processing. This option may nevertheless be left unchecked for compatibility reasons. Keep in mind that you must restart the database after modifying this option in order for the new operation to become effective.

Options

- **Use Javascript for Entry Control**: When this option is checked, part of the browser's entry control is taken over by automatic Java scripts. The data entry controls and the data types (fields or variables) to which they can be applied are as follows:
  - minimum value (for numeric values);
  - maximum value (for numeric values);
  - mandatory value (for numeric and alphanumeric values).

1. A space character must be inserted between 4DVAR and the variable name.

Generated Java scripts, which are small in size, display alert dialog boxes during data entry without preventing the user from validating the entry (which is still 4th Dimension's responsibility):

- **Save Request in File (logweb.txt)**: This option lets you generate a log of requests sent to the Web server of 4th Dimension. The log is presented in the form of a text file named "logweb.txt" automatically placed at the same level as the structure file of the database. This file is in CLF (Common Log Format) or NCSA format, recognized by most Web site analysis tools.

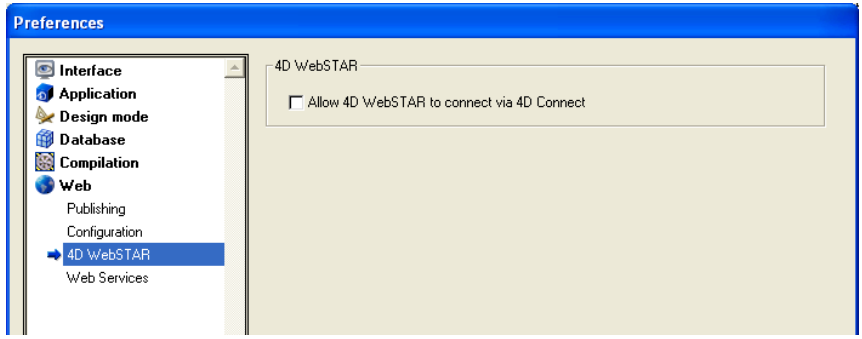
  Each line of the file represents a request, such as:
  host rfc931 user [DD/MMM/YYYY:HH:MM:SS] "request" state length.

  Each field is separated by a space and each line ends by the CR/LF sequence (character 13, character 10).

  For more information about this file, refer to the "Information about the Web Site" section of the *4th Dimension Language Reference* manual.

4D WebSTAR Page

This page lets you set the access of a 4D WebSTAR server to the 4th Dimension Web server:



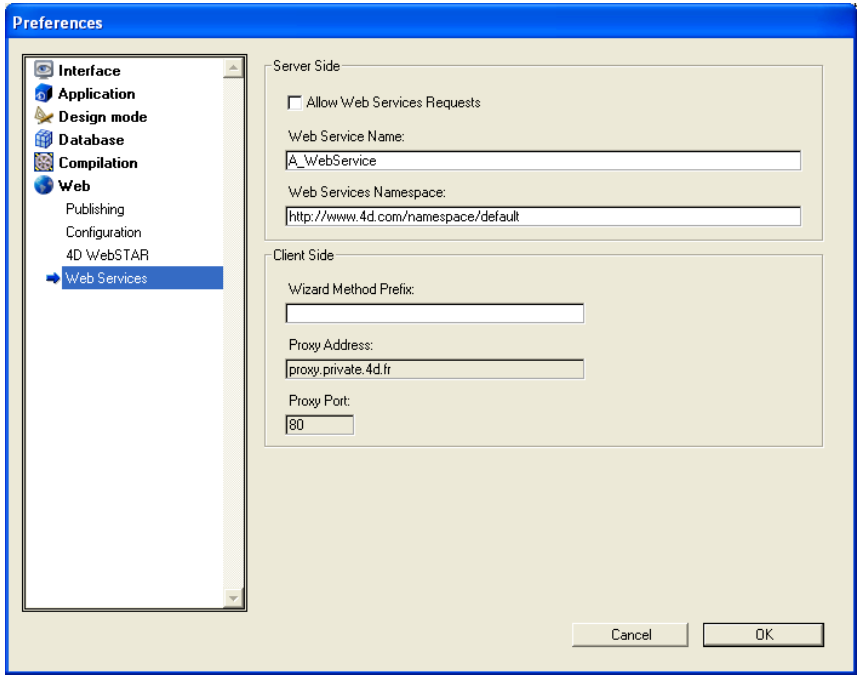
- **Allow 4D WebSTAR to connect via 4D Connect**: This option is designed to allow (checked) or disallow (unchecked) 4D Connect plug-in connections to the 4D Web server.

4D Connect is a plug-in for the 4D WebSTAR Web server, allowing it to communicate with a 4th Dimension Web server.
For security reasons, this option is not checked by default. Depending on your Web configuration, 4D, Inc. recommends the following settings:

- If your 4th Dimension Web server is not connected to a 4D WebSTAR server using the 4D Connect plug-in, leave this option unchecked.
- If your 4th Dimension Web server is connected to a 4D WebSTAR server using the 4D Connect plug-in, you must check this option for it to work.
  In this configuration, it is recommended to run the 4th Dimension Web server behind a firewall and to filter, using this firewall, requests addressed to 4D.

Web Services Page

This page is used to define the general parameters concerning the publication and use of *Web Services* in your 4th Dimension database. For more information about Web Services, refer to the chapter "Publishing or Using Web Services," page 723.



Server Side

This area contains various options related to the use of 4th Dimension as a Web Services "server," i.e., publishing project methods in the form of Web Services.

- **Allow Web Services Requests**: This option lets you "initialize" the publication of Web Services. If this option has not been checked, 4th Dimension refuses SOAP requests and does not generate a WSDL — even if the methods have the **Offered as a Web Service** attribute. When this option is checked, 4th Dimension creates the WSDL file.
- **Web Service Name**: This area lets you change the "generic name" of the Web Service. This name is used to differentiate the services both at the SOAP server level (when the server publishes several different Web Services), as well as in the Web Services directories. By default, 4th Dimension uses the name A_WebService.
- **Web Services Namespace**: This area is used to change the namespace of the Web Services published by 4th Dimension.
  Each Web Service published on the Internet must be unique. The uniqueness of the names of Web Services is ensured using XML namespaces. A namespace is an arbitrary character string used to identify a set of XML tags in a unique way. Typically, the namespace begins with the URL of the company (http://mycompany.com/mynamespace). In this case, it is not indispensable to have anything in particular at the defined URL; what matters is that the character string used is unique. By default, 4th Dimension uses the following namespace: http://www.4d.com/namespace/default.

Note In conformity with the XML standard for tag names, the character strings used must not contain spaces or extended characters. Only the following Latin characters can be used: ([A-Za-z0-9._] | '-')*.

Client Side

This area contains various options related to the use of 4th Dimension as a Web Services "client," i.e., subscribing to services published on the network.

- **Wizard Method Prefix**: This area lets you change the prefix that is added automatically by 4th Dimension to the name of proxy methods generated by the Web Services Wizard. Proxy project methods form a link between the 4th Dimension application and the Web Services server.
  By default, 4th Dimension uses the prefix "proxy_".
- **Proxy Address** and **Proxy Port**: This area lets you check the current Internet connection parameters, used for the subscription to Web Services. These values are only read by 4th Dimension; if you want to modify them, you must do so via the Internet parameters of the machine.

3 Designing a Database Structure

This chapter tells you how to use 4th Dimension's Structure editor to create and modify database structures. The structure of a database consists of tables and fields. If a database has more than one table, the structure might include the relations between the tables.

The structure of a database is like the foundation of a house — it provides the basis for everything else. This chapter gives a general description of databases and of how to design database structures to meet different information management needs.

This chapter tells you how to:

- Manipulate table images in the Structure editor,
- Create tables and set table properties,
- Create fields and set field types and properties,
- Relate tables.

Database Basics

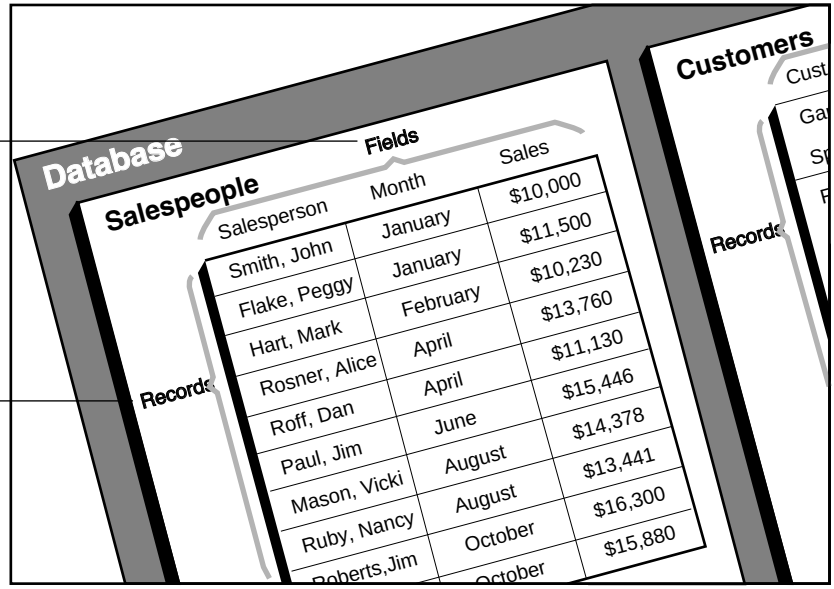
A database is any collection of information organized so that it can be used efficiently. A telephone directory is a good example of a database. So is a dictionary, calendar, or cookbook.

The information in a database is organized in the form of records. Each record contains all of the information about one person or thing in the database. For example, each record in a telephone directory contains one person's name, address, and telephone number.

Each record contains fields. A field is used to store a particular piece of information. For example, in the telephone directory database, one field contains the person's name; a second field contains the person's address and a third field contains the person's telephone number. Every record contains each of these fields and every record can have information in these fields.

A field name usually identifies the information that goes into the field. A field name is usually something like Name, Address, or Phone Number. Each field has a field type that identifies the kind of information that can be entered in a field: numbers, dates, alphanumeric characters, and others. Because each field contains a specific type of data, you can perform calculations and other operations on the information in the fields. For example, numbers from two fields can be added. A date in one field can be compared to a date in another field. A person's first name (stored in one field) can be displayed in front of the last name (stored in another field) to make the first line of an address label.

All the records together make up a *table*. Each database can contain many tables. The following figure shows how these concepts are related.



4th Dimension can reorganize records and perform calculations on the information so that the information is useful. For example, 4th Dimension can calculate the total values in a field and present the total in a report. It can calculate a total for each salesperson and display a graph that compares sales figures.

Tables

When you create a new database, 4th Dimension automatically creates one table and names it *Table1*. You can rename the table and add fields to it. To use the database, you must add at least one field.
4th Dimension allows you add additional tables to the structure and establish relations among tables. This gives you the ability to create a structure that meets your exact needs.

Single-Table Structures

Some databases use only one table. You use a single table for a single category of information such as people, companies, or inventory. You can have as many fields in a table as you need (up to 511). Non-relational database applications sometimes refer to a single table database as a *flat-file database*.

Table image in the Design environment

Records in the User environment

First Name	Last Name	Address	Phone
Biff	Davis	689 Elm St.	758-3652
Shirley	Ransome	9087 Ridge Road	252-6892
Lance	Wolfram	333 Main Street	895-6686
Dennis	Hanson	4567 Remmington Pl.	456-9256
Lydia	Vernon	978 Ortega St.	682-6983
Andy	Venable	10098 Oregon Rd.	563-3654

In the figure above, every person's record needs the same types of data. The database grows in accordance with the number of people stored.

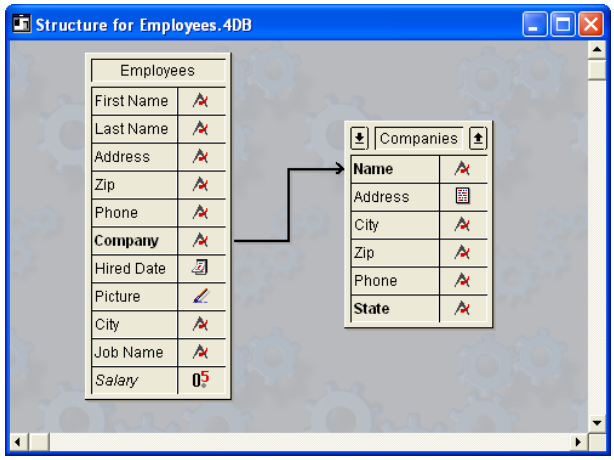
Multiple-Table Structures

A database can often store and access data more efficiently by using more than one table. A good rule to remember is that different types of information should be stored in different tables.

A database that keeps track of both people and companies is a good example. The records for the people and the companies are stored in different tables. If the address of a company changes, you need only change that company's record. You do not need to enter that new address for every person who works for the company.

With a single table, you would have needed to enter the address in each individual record; with two tables, you need to enter that information only once. When a company name is entered in a person's record, 4th Dimension can search for the company's record and automatically display the correct address.

The figure below shows the structure of a multiple-table database in which two tables are related. The arrow drawn between the Company field and the Company Name field shows that relationship.



The data for each person is stored in the [People] table. Data about each company is stored in the separate [Companies] table[1].

4th Dimension is called a *relational database* application because it can use multiple tables and relate them in various ways. For example, you can create a report for the [People] table that searches the [Companies] table and automatically displays and prints information about each person's company. The relationship between the tables allows information from each table to be available to the report.

You can also enter data directly into related tables. For example, an invoicing database can write information to a [Line Items] table from within an Invoicing screen. You can also write data to related tables using 4th Dimension's language.

For information on creating and using related tables, see the section "Relating Tables," page 193.

Sometimes you need a multiple-table structure in which tables are not directly related. It may be convenient to have one database store different kinds of information such as a contact list and an expense table.

1. In the 4th Dimension documentation, table names are shown in brackets. This is how they appear in the Method editor.

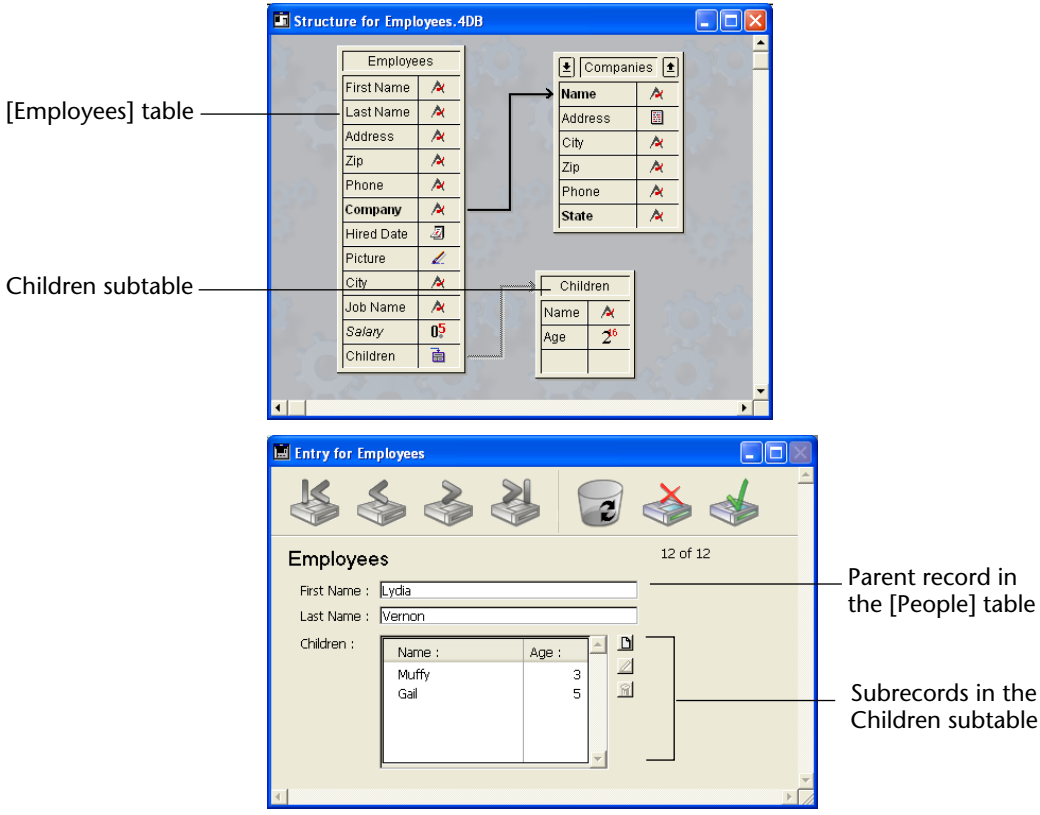
4th Dimension allows up to 255 tables in each database. A table can have up to 511 fields. Using multiple tables, virtually any kind of database structure is possible.

Subtables

Occasionally you will need to store a variable amount of information for each record in a table. 4th Dimension allows you to create a table structure that includes a subtable — a table subordinate to a parent table. Information in the subtable is available only when the parent record is in use.

For example, suppose you want to keep track of the children of the people in your table. Some people have no children and some people have many children. You can use a subtable to store this information.

The figure below shows a subtable and form that displays subrecords belonging to the parent record.



This subtable allows you to create as many subrecords as you need for each record in the [People] table. If a person has three children, you create three subrecords for that record. If a person has no children, you create no subrecords.

A subtable structure is useful when you need to see and use the subrecords *only* when its parent record is being used. For example, using subtables you could easily find the average age of one person's children, but it would be difficult to find the average age of everyone's children. Similarly, it would be difficult to do a search across all children in the whole database (e.g., to find all five-year-old children). If performing these kinds of operations on the data is a requirement, it would be better to use a separate table for children, not a subtable.

Note Whenever a parent record is loaded, 4th Dimension loads all the subrecords belonging to that parent record. If there are many subrecords belonging to the parent record, this can be undesirable. In this case, it is recommended that you use a related table to store the information and that you control the loading of records using the language.

You cannot access information contained in a subtable from a 4D Open application. 4D Open is the API (Application Programming Interface) that allows non-4D Client applications to connect to 4D Server.

Creating a Database Structure

Every database has a structure that consists of at least one table and at least one field. These elements must exist before the database can store records.

▶ Typically, you create a structure with the following basic steps:

1 **Create a new database.**

For more information, see "Creating a New Database" on page 24.

4th Dimension automatically creates the first table.

2 **Rename the automatically created table (optional) and assign table properties.**

For more information, see "Renaming a Table" on page 166 and "Setting Table Properties" on page 165.

3 **Create fields for the table and assign field properties.**
For more information, see "Creating New Fields" on page 172.

4 **Add additional tables and fields as necessary.**
For more information, see "Creating a New Table" on page 164.

5 **Relate one table to another, if necessary, by drawing a line between fields in the two tables and setting relation properties.**
For more information, see "Relating Tables" on page 193.

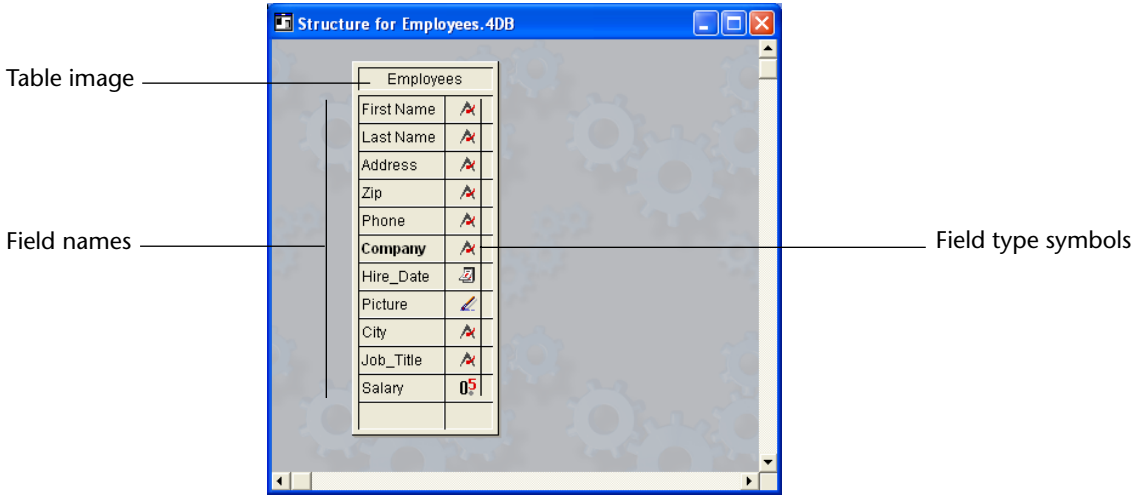
The remainder of this chapter describes these steps in detail.

Using the Structure Editor

The Structure editor lets you manage the database structure — the tables and their relationships. It gives you control over such things as tables, table properties, fields, field properties, and table relations.

The Structure editor gives you a graphic view of a database's structure and provides menus for performing design operations. Each table is represented by a table image in the Structure editor. It shows the fields and their types, in the form of icons or letters. 4th Dimension automatically creates the first table. You can add tables as necessary.

The following illustration shows the Structure editor window with one table image.



Selecting a Table Image

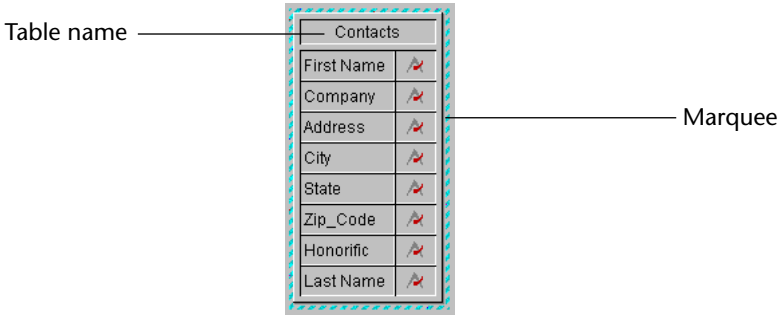
To work with a table in the Structure editor window, you first need to select it[1]. You can then move, or resize the table image.

► To select a table in the Structure editor.

1 **Click the image of the table**
OR
Right-click (Windows) an empty area of the Structure editor window or press the Ctrl key while clicking an empty area of the Structure editor window (MacOS), then select the table from the Table List submenu of the contextual menu.
OR
Press the Tab key to select through the tables and subtables of the structures.
4th Dimension draws a flashing marquee around the table image and makes it the frontmost table[2]. Subsequent actions affect the selected table.

Note Unless a table is selected, the Table Properties and New Field menu commands in the Structure menu are disabled.

The figure below shows a selected table.

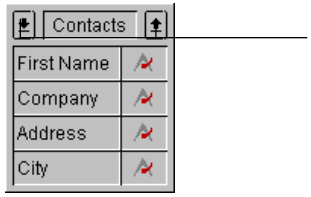


Scrolling the Field List

As you add fields to a table, you may add more fields than are visible in the table image displayed in the Structure editor window. When this happens, 4th Dimension automatically adds scroll arrows to the title area of the table image.

1. If the desired table image is not in view, you can double-click its name in the Tables page of the Explorer to bring it into view.
2. When tables are superposed in the Structure editor window, their position determines the selection order when you navigate through the tables using the Tab key. 4D stores this position for each user between two sessions.

You can scroll the field list within the table image by clicking on a scroll arrow.



Resizing a Table Image

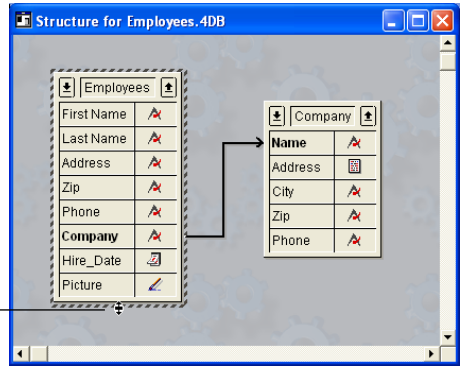
You can also expand a table image to show more fields in the table.

► To expand a table image:

1 **Position the pointer at the bottom edge of the table image until it changes to a table resizing pointer ↕.**

2 **Drag the bottom edge down.**

The following illustration shows a table image being resized.

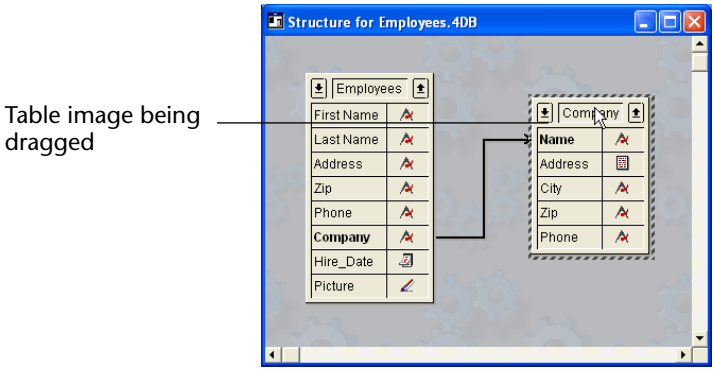


As you drag, the table image expands in increments of one field at a time. When all fields become visible, the scroll arrows in the title bar of the table image automatically disappear.

4D Server If you resize a table image when using 4D Server, the table appears resized for all users in the Design environment.

Moving a Table Image

As you add tables to your structure, you may need to move the table images around to make room or reorganize the table images. You can do so by dragging the table name area. The illustration below shows a table image being dragged.



- **To move a table image, drag the table name bar.**

  Drag the table name bar only. Dragging other parts of the table image may produce different effects, such as creating a new table relation or changing the size of the table image.

  If the table image that you move is related to another table or subtable, the connecting lines move correspondingly, maintaining their attachment to the other table or subtable.

4D Server If you move a table image when using 4D Server, the table appears in its new location for all users in the Design environment.

Creating a New Table

When you create a new database, 4th Dimension automatically creates the first table in the database. You can create additional tables at any time.

4th Dimension names the first table [Table1]. 4th Dimension names additional tables sequentially, up to [Table255]. You can rename the tables at any time. See "Renaming a Table" on page 166 for more information about naming tables.

You cannot delete tables. However, if you create an unwanted table, you can make it invisible to users (for more information, see the section "Setting Table Properties" on page 165).

► To create a new table:

1 **Choose New Table from the Structure menu.**
OR
Press Ctrl+N (Windows) or Command–N (MacOS).
OR
Click the New Table button in the toolbar  **.**
OR
Right-click (Windows) an empty area of the Structure editor window or press the Ctrl key while clicking an empty area of the Structure editor window (MacOS), then choose New table from the contextual menu.

The following dialog box appears.

If you want to cancel the operation, click the **Cancel** button.

2 **Enter the name of the table you want to create and click the OK button.**

You can enter up to 31 characters in the name entry area. The name can include any combination of characters, dashes, numbers and underscores.

4th Dimension removes any character past the 31st character and also removes spaces that are located at the beginning or the end of the name.
4th Dimension creates a new table image. It becomes the selected table image in the Structure editor window.

3 **Repeat steps 1 and 2 for each table you want to add to the database.**

Setting Table Properties

You use the Table Properties window to set several properties of each table. With the Table Properties window, you can:

- Rename a table,
- Set access privileges,
- Specify the events for which the table's trigger is active,
- Customize the appearance of the table image in the Structure editor window.

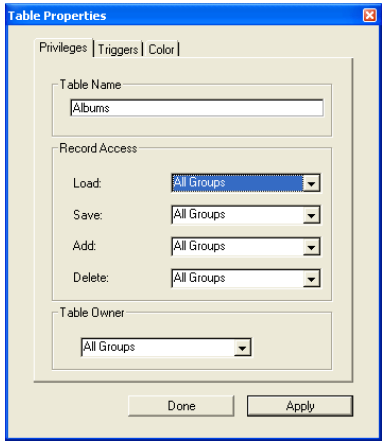
Displaying the Table Properties Window

The Table Properties window displays the properties of the selected table. If it is not displayed, you first need to open it.

▶ If the Table Properties window is not displayed:

1 **Click on the table image and choose Table Properties from the Structure menu.**
OR
Double-click the title bar of the table image.
OR
Right-click (Windows) the title bar of the table or press the Ctrl key while clicking the title bar of the table (MacOS), then select Table Properties from the contextual menu.
OR
In the Explorer, double-click on the table name or image on the Tables page, or select the table then click on the Edit button.

4th Dimension displays the Table Properties window, showing the properties for the selected table.



Note You can also display the Table Properties window by double-clicking the table name in the Tables page of the Explorer or highlighting the table name and clicking **Edit**.

► If the Table Properties window is displayed but shows the properties of another table:

1 Click on the table image whose properties you want to set.

For more information about selecting a table image, refer to "Selecting a Table Image" on page 161. Once you have selected the table image, the Table Properties window reflects the properties of the newly selected table.

Renaming a Table

You will usually want to rename a table so that the table name identifies the information it will contain. For example, if [Table1] will contain customer records, you might rename it [Customers].

You can rename tables at any time. If you have used the old table name in a method, 4th Dimension automatically changes it to the new name *provided the method is closed*. If the method is open, you must make the changes yourself.

Note Do not use the same name for two tables in the same database. If you inadvertently create a duplicate table name, 4th Dimension recognizes only the table you created first, ignoring any tables subsequently created with the same name. If you attempt to give the same name to two different tables, an alert is displayed that allows you to cancel the name entry.

- **To rename a table, enter a name in the Table Name area of the Table Properties window.**

  You can enter up to 31 characters in the Name area. The table name must begin with a letter. The remainder of the table name can contain any combination of letters, numbers, spaces, and underscores.

  4th Dimension truncates table names longer than 31 characters and removes spaces at the beginning or end of the name.

Setting Record and Table Access Privileges

The drop-down lists in the Record Access area allow you to assign groups access to various operations in the User and Custom Menus environments. Members of the group assigned Load privileges can view records in the table, members of the group assigned Add privileges can add records to the table, and so on.

The **Owner** drop-down list in the Table Access area allows you to give a group the ability to modify the table properties in the Design environment.

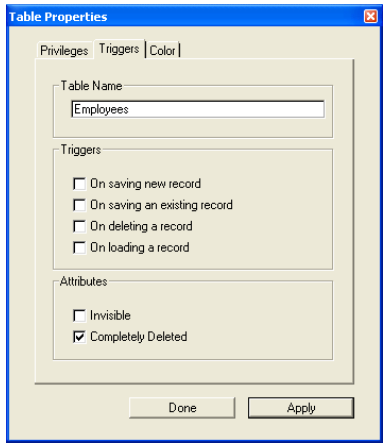
For complete information about creating a password access system and assigning access privileges, see the chapter "Managing Password Access," page 613.

Setting Trigger Events

A trigger is a method that runs automatically when certain events related to the table occur. The events are:

- Saving a new record,
- Saving a record,

- Deleting a record,
- Loading a record.



Triggers are created using the Methods page of the Explorer. You can access this page directly from the Structure window using a **right-click** (under Windows) or **Control+click** (under MacOS) on the title bar of the table image then selecting **Show Table Form Methods & Triggers...** in the contextual menu.
On the Triggers page of the Table Properties window, you can specify the events that will execute the trigger of a table. For more information about creating and editing triggers, see the section “Triggers,” page 519.

► To set the events for which the trigger is active:

- **Click one or more trigger check boxes.**

  The trigger you enter in the Method editor will run only when the events you check occur.

Note In the “Compatibility” area of the **Design Mode/Options** page of Preferences, you can choose to execute triggers according to the rules for File procedures in previous releases of 4th Dimension. For more information, see the section “Options Page,” page 127.

Setting Table Attributes You can set the following two table attributes:

- **Invisible table** This check box allows you to make a table invisible in the User and Custom Menus environments. Use this option when a table is no longer being used in the database.

Note The Invisible property can also be set using the contextual menu that appears when you right-click the title bar of the table image (Windows) or when you press the **Ctrl** key while clicking the title bar of the table image (MacOS).

Making a table invisible allows you to limit the operations that a user can perform on a table by making the table and its fields invisible in all editors and some dialog boxes that appear in the User and Custom Menus environments.

These editors and dialog boxes include the following:

- All query editors
- Order By editor
- Chart editor
- Label editor
- Quick Report editor
- Import and Export dialog boxes
- Apply Formula dialog box

In each of these editors, the user is unable to see or choose the table or any of its fields. For instance, the user cannot include any fields from an invisible table in a report or label.

When using the editors, users have the option of saving their specifications (e.g., the query or sort they created) to disk files. In this case, any invisible tables or fields specified will be used in the operation. In addition, users can type the names of invisible tables and fields in the Apply Formula dialog box.

In the Design environment, invisible tables and fields are displayed in italics in the Structure editor window.

4D Server Object locking occurs when more than one person tries to modify the same table's properties at the same time. If a user is modifying a table's properties, the properties are locked and cannot be modified by other users. The properties remain locked until the first user finishes modifying the properties and closes the dialog box (i.e., by clicking the **Cancel** button).

- **Completely Deleted** This option in allows you to optimize the deletion of a selection of records that are deleted with the DELETE SELECTION command.

  When 4th Dimension deletes a selection, the tags of the deleted records are also cleared. A tag is a header attached to a record that carries information about that record.

  When both records and tags are deleted, the process is significantly slower than only deleting the records. Therefore, you may not always wish to have the tags cleared automatically. You can choose whether or not this action is automatic.

  To speed up the deletion of a large selection of records with the DELETE SELECTION command, uncheck the **Completely Deleted** check box. Deselecting this option tells 4th Dimension not to clear tags when it deletes. This option cannot be set using the language.

By default, 4th Dimension deletes records and tags. If you deselect this option, 4th Dimension will not delete the record tags. Therefore, if you perform a recover by tags, you will recover the deleted records or whatever has been put in the holes corresponding to the deleted records. For this reason, it is recommended that you deselect this option only for tables that are not important, such as temporary tables. If you deselect this option and you must repair by tags, you can avoid recovering the deleted records by compacting the data table before performing the recover by tags (see the documentation for 4D Tools for information about compacting).

Setting the Color of the Table Image

The third page of the Table Properties window lets you set the color of either the table name or the background of the table image in the Structure editor. You can indicate in the Preferences dialog box whether the color you choose applies to the name or to the background. For more information, see the section "Structure Editor Page," page 125.

The Color property lets you use color to help organize the structure of a large database. For example, you could use one color for all tables that relate to customers and customer records another color for tables that relate to inventory and inventory records.

► To set a color:

1 **In the Color page of the Table Properties dialog box, click the desired color in the Color palette.**

When you click **Apply,** either the table name or the table background changes to the color you selected (depending on the Preference you selected).

OR
Right-click (Windows) the title bar of the table or press the Ctrl key while clicking the title bar of the table (MacOS); select Color in the contextual menu then choose a color from the Color palette.

The color is applied immediately.

► To set the color back to the default color.

1 **In the Color page of the Table Properties dialog box, select the Default Color options.**

The default color is applied when you either click the **Apply** or **Done** button.

Creating Fields and Setting Field Properties

For each table, you need to create the fields that hold the data you want to store and manage.

When you create a field, you assign it a field type that describes the kind of information that will be stored in the field. 4th Dimension uses the field type to perform different kinds of operations on the contents of the field. For example, if a field will contain a date, you will want to create it with a Date field type. Subsequently, 4th Dimension can compute date values, such as length of service or qualification for benefits. In addition, 4th Dimension can sort records in chronological order using the dates in this field. Field types are described in detail in the section "Field Types" on page 176.

In addition to the field type, each field in a table can possess any of six attributes. Attributes determine conditions for entering, displaying, or modifying data in the fields. They are described in the section "Field Attributes" on page 182.

With the exception of Subtable fields, after you create a field, you can return to it to change the field type or any of its attributes.

Creating New Fields

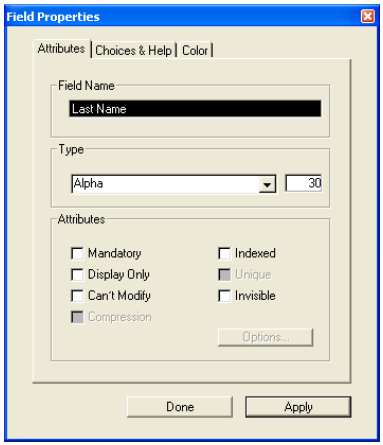
Each time you add a new field to a table, you:

- Name the field,
- Specify the field type,
- Set field properties (optional).

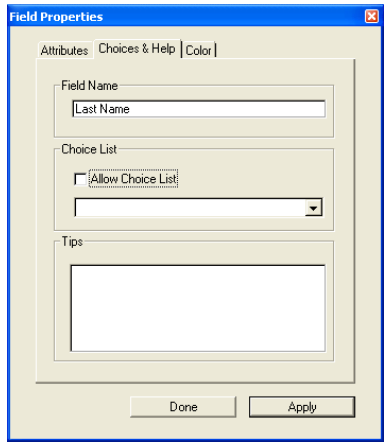
You can add up to 511 fields to a table or subtable. 4th Dimension adds fields to the table in the order that you create them. You cannot reorder or delete fields. You can, however, order the fields any way you want in the input and output forms you will use to enter and manage the data.

Refer to the chapter "Creating Forms," page 225 for information about creating forms.

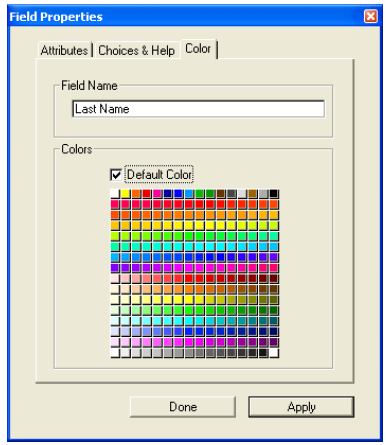
You create new fields and set field properties using the Field Properties window. The Field Properties window has three pages: Attributes, Choices & Help, and Color. The pages are shown in the following illustration.



Set field name, type, and attributes.



Assign a choice list and enter a help message.

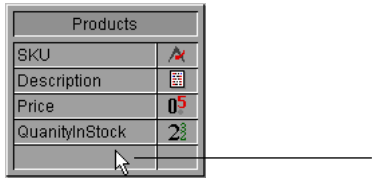


Set the field color in the table image.

You cannot delete fields. However, if you create an unwanted field, you can make it invisible to users by choosing the Invisible attribute. For more information, refer to "Field Attributes" on page 182.

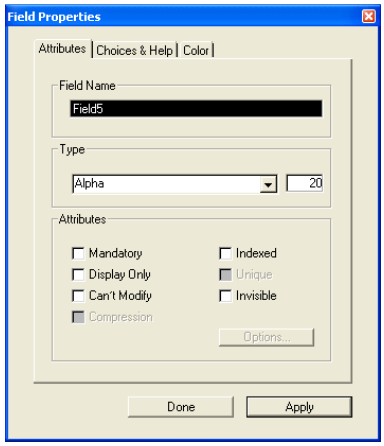
► To create a field:

1 Select the table image in which you want to create a field in the Structure editor and select New field from the Structure menu (or click on the New field button in the toolbar).
OR
Double-click an empty row in the table image below the existing field names.



OR
Right-click (Windows) the title bar of the table or press the Ctrl key while clicking the title bar of the table (MacOS), then select New field from the contextual menu.

4th Dimension displays the Attributes page of the Field Properties window.



2 Type the field name in the Field Name area.

You can enter up to 31 characters in the Field Name area. The field name must begin with a letter. You can then use any letters or numbers, as well as spaces and underscores. 4th Dimension truncates field names longer than 31 characters and removes spaces at the beginning or end of the name.

Do not use the same name for two fields in the same table. If you inadvertently create a duplicate field, 4th Dimension displays an alert that prevents you from reusing same field name in a table. Do not leave a field name blank or use any reserved words in the field name. Reserved words include command names (e.g. Date, Time, etc.), keywords (e.g. If, For, etc.) and constants.

Tip Although you can include spaces in field names, entering a field name that does not include any spaces allows you to double-click to select the field name in the Method editor. For this reason, it is generally advisable to use underscores instead of spaces.

3 Select a field type and, if necessary, modify the maximum field length.

For more information on field types and field lengths, see the section "Field Types" on page 176.

4 Select any attributes (optional).

For more information, see the section "Field Attributes" on page 182.

5 Click the Choices & Help tab and enter a help message for the field or assign a choice list (optional).

Balloon Help can be viewed only when the database is used on a Macintosh. A tip is displayed on all platforms. For more information, see the section "Help" on page 188.

A choice list is displayed automatically when the field is selected for data entry or when the field is used in the Query editor. A choice list can be assigned in the Field Properties window or can be assigned on a form-by-form basis in the Form editor. For more information, see the section "Using Choice Lists" on page 374.

6 **Click the Color tab to assign a color to the field (optional).**

The field color can be used in the Structure editor to distinguish fields. For example, you can use a color to highlight the field that uniquely identifies each record. The color is applied to either the text of the field name or the background rectangle containing the field name, depending on the Color Field Names or Background preferences. For more information, see the section "Setting the Color of the Field" on page 189.

7 **When you are finished assigning properties, click the Apply button to save the field properties.**

You do not have to click **Apply** separately on each page. Clicking Apply once after you have entered all your field properties saves the properties on all three pages.

When you click **Apply** after creating a new field, 4th Dimension automatically displays default properties for another new field. The default name is *FieldN*, where N is the sequence number of the field in the table. If you want to create the new field, modify the default properties.

When you are finished creating fields, click **Done**. When you click **Done**, 4th Dimension puts away the Field Properties window.

With the Field Properties window on-screen, you can modify the properties of any existing field simply by clicking the field in its table image or double-clicking its name or its preview image on the Tables page of the Explorer. You can also cycle through tables by clicking one table image and pressing the **Tab** key. You can navigate through the fields using the following keys.

- **Up** and **Down** arrow keys: to go up and down in the field list of the selected table.
- **Home** and **End** keys: to go at the top or bottom of the field list of the selected table.

- **PgUP** and **PgDn**: to display the next or previous sequence of fields in the selected table image.

4D Server Object locking occurs when more than one user tries to modify the same field definition at the same time. If a user is modifying a field's properties, the properties are locked and cannot be modified by other users until the first user unlocks the field properties by clicking the **Done** button).
In addition, while the field properties are locked, other users cannot modify the properties of other fields in that table. Other users can, however, modify the table properties and the properties of other fields and tables in the database.

Field Types

You must specify a field type for each field. Field types affect how 4th Dimension manipulates and stores data in a field and how you enter or display data in forms. 4th Dimension supports the following field types:

- **Alpha** Alphanumeric text between 2 and 80 characters
- **Text** Text up to 32,000 characters
- **Real** Floating point number in the range of 1.9E-4951 to 1.1E4932
- **Integer** Number between -32,768 and 32,767
- **Long integer** Number in the range of plus or minus 2,147,483,647
- **Date** Date between the year 100 and the year 32,767
- **Time** Time in hours:minutes:seconds format
- **Boolean** A field that can only take the values TRUE or FALSE
- **Picture** A PICT image
- **BLOB** Any binary object such as a graphic, another application, or any document

You set the field type in the Field Properties window. Here are complete descriptions of each field type.

Alpha

An Alpha field contains alphanumeric characters (letters and numbers), punctuation marks, and special characters such as the asterisk (*), percent sign (%), hyphen (-), and so on. Use an Alpha field to contain any information that must be treated as text and does not exceed 80 characters in length.

An Alpha field can be indexed (whereas a Text field cannot be indexed). For information about indexing, see the section "Field Attributes," page 182.

Alpha is the most common field type. Typically, you use this field type for names, addresses, telephone numbers, postal codes, and so forth. During data entry, an Alpha field accepts any character, number, punctuation mark, or special character.

Zip codes are best placed in an Alpha field for two reasons: Numeric fields do not display leading zeros and some zip codes contain a hyphen. The general rule for deciding between a numeric field type or an alphanumeric field type is make it an alphanumeric field unless it will be used in a numerical calculation or searched or sorted based on numeric values.

You can set the maximum length of an Alpha field to be between 2 and 80 characters long. For example, if you use a field for state abbreviations, you could limit it to 2 characters.

You can concatenate two or more Alpha fields. For instance, you might want to join a person's first name and last name for the first line in a label form.

You can do so using a one-line method, such as:

```
FullName:=First_Name+" "+Last_Name
```

The variable *FullName* can be displayed or printed. You can also extract part of the information for use in another place (extraction of a substring). The substring can be displayed or printed.

Text

A Text field can hold up to 32,000 alphanumeric characters. You use a Text field to hold blocks of text longer than 80 characters such as comments or descriptions.

In an input form, a Text field can be given a vertical scroll bar. In a printed report, the Text field area can expand as necessary to print all the information, even if it covers several pages.

During data entry, Text fields provide basic text editing features: scrolling, word wrapping within the area set for the field display, double-clicking to select a word, moving the insertion point with the arrow keys, and standard cut, copy, and paste operations. A Text field accepts a carriage return during data entry to create a new paragraph (an Alpha field does not).

You can paste text into Text fields, including text from word processors.

You cannot index a Text field, but you can perform a search based on characters in the field.

Another way to store text with a record is to use the 4D Write plug-in. With 4D Write, you can use different font attributes, paragraph alignments, and other word processing features that are not available in standard Text fields. For more information about using 4D Write, refer to the documentation that comes with 4D Write.

Real

A Real field stores real numbers, that is, decimal numbers (price, salary, expenses, and so on). Real number fields can hold any number in the range of 1.9E-4951 to 1.1E4932.

Note In the United States, the decimal separator in real numbers is the decimal point (.). In other countries, it is often a different character such as the comma (,). 4th Dimension uses the decimal separator specified in the operating system of your machine.

Integer

Use an Integer field type for any field that stores whole numbers, that is, numbers without decimals (record number, invoice number, and so on). Integer fields can contain whole numbers between -32,768 and 32,767.

Long Integer

Use a Long Integer field type for any field that stores whole numbers that are too large for an Integer field. They can contain whole numbers (no decimal) between ±2,147,483,647.

Date

Use a Date field to store date values such as Start Date, Date Purchased, Birthdate, and so on. A Date field can store any date value (month, day, year) entered in a *MM/DD/YYYY* format between the year 100 and the year 32,767.

Note In the United States, dates are specified in the month/day/year (*MM/DD/YYYY)* format. Other countries use different formats such as *DD/MM/YYYY* for British systems and *YY/MM/DD* for Swedish systems. 4th Dimension will store the date based on the format specified by the operating system of your computer.

Time

Use a Time field type to manage times such as Current Time, Meeting Time, Billed Time, and so on. A Time field can store any time value entered in *HH:MM:SS* format.

Boolean

Boolean fields (sometimes called logical fields) contain TRUE or FALSE values.

You can format a Boolean field as either a check box or as a pair of radio buttons. A check box that contains a check is TRUE; empty, it is FALSE. Either the first radio button is selected (TRUE), or the second button is selected (FALSE).

You should name a Boolean field so that you can ask the question, "Is *field name* true?" This question is useful for searching because during a search, 4th Dimension looks for a TRUE and FALSE value in a Boolean field. For example, you might want to name a field "Male" instead of "Sex." Your search condition can then be written "Male is equal to true," instead of "Sex is equal to true."

Picture

Picture fields are used for several purposes in 4th Dimension. Creating a Picture field allows you to save the following types of data.

- **Pictures** You can store digitized photographs, diagrams, maps, and illustrations created using a graphics application. Some graphic applications store extra information with pictures that may provide special instructions for output devices such as a PostScript™ printer. This information "tags along" when the picture is copied or pasted into a Picture field and is used by 4th Dimension when printing the picture to an appropriate output device.

- **Data from 4th Dimension plug-ins** You can store data created with some of the 4th Dimension plug-ins — (4D Chart, 4D View, 4D Write, etc.). For more information about using a 4th Dimension plug-in a Picture field, refer to the documentation that comes with the plug-in.
- **Data from third-party plug-ins** You can also store data from third-party plug-ins (if the plug-in supports saving data). For more information about storing data from a plug-in in a Picture field, refer to the documentation included with the 4th Dimension Plug-ins Kit.

Note Data that comes from third-party plug-ins can also be stored in BLOB fields. For more information on this feature, refer to the documentation that comes with the plug-in.

BLOB

Blob (**B**inary **L**arge **Ob**ject) fields store binary documents of any kind. For example, you can store documents created by other applications, scanned images, or other applications. A BLOB can be as large as 2 gigabytes. When you are working with a record that contains a BLOB field, the entire BLOB is loaded into memory. You can use a BLOB field to store entire desktop documents within your database. You can also write the contents of a BLOB field to a desktop document. For example, you can use a BLOB field in a document management system that stores documents in the database and delivers them to users upon request.

You use Blob commands in 4th Dimension's language to manage BLOB fields. Use the DOCUMENT TO BLOB and BLOB TO DOCUMENT commands to read and write documents to and from BLOB fields. The commands COMPRESS BLOB, EXPAND BLOB, and BLOB PROPERTIES let you work with compressed BLOBs. For more information about working with BLOBs, see the section on BLOBs in the *4th Dimension Language Reference* manual.

The contents of a BLOB field is not displayed on-screen since a BLOB can represent any type of data.

Subtable

A Subtable field is a field that associates a subtable with each individual record in a table. For example, a [People] table could include Children as a Subtable field. The subtable associated with the field — also called Children — could contain fields that store data on each person's children (their names, ages, birth dates, and so on). A single subtable can contain up to 32,767 subrecords and each subrecord can have as many as 511 subfields.

The record to which a subtable is attached is called the parent record and the table that contains the parent record is called the parent table. Each subtable has its own set of fields, called subfields.

You can use subtables to manage a variable number of subrecords. For example, you may want to create a student table in which each record contains data about the student's honors and the dates of the awards. Instead of having to create a number of fields such as Honor 1, Honor 2, and so on, you can create a subtable to which you can add subrecords as the awards accumulate.

In many cases, however, it is best to use a related table instead of a subtable. Subtables have several limitations that related tables do not. First, a subtable cannot be viewed without opening the parent record. Second, it is difficult to use information across subrecords. For instance, you could easily produce a sorted list of each student's honors but it would be difficult to produce a sorted list of all honors awarded to all students. If you need to generate this type of sorted list, you should use a related table.

In general, you should not use a subtable to store information that you will need to search on, access directly, or use for calculations.

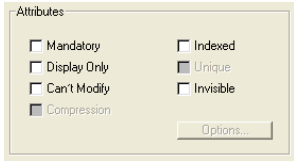
Because 4th Dimension loads subtables into memory when it loads their parent records, the number of subtables and subrecords is limited by the amount of available memory. A good rule of thumb is to allow no more than 100 subrecords per parent record. If processing speed is of concern, allow no more than 25.

You add subfields to a subtable the same way that you add fields to tables. You cannot create more than one level of subfields.

After you define a field as a Subtable field, you cannot assign it another field type. However, you can make the field invisible in the User or Custom Menus environment (for more information, see the section "Invisible" on page 185).

Field Attributes

Field attributes determine conditions for entering, displaying, or modifying data in the field. Each field can have several attributes.



You set the field attributes in the Attributes page of the Field Properties window. Attributes that cannot be selected for a particular field type are disabled. A description of each of the field attributes follows.

Note The **Invisible** and **Indexed** attributes can also be set for a field using the contextual menu that appears when either right-clicking a field in the Structure editor (Windows) or pressing the **Ctrl** key when clicking the field in the Structure editor window (MacOS).

Mandatory

When the Mandatory attribute is set for a field, the user must enter a value in that field during data entry. 4th Dimension does not accept a record that contains an empty mandatory field. You would set the Mandatory attribute for a field that contains essential information for your database. The field that uniquely identifies each record is a good candidate for the Mandatory attribute. Social Security numbers, invoice numbers, certain dates, or employee numbers might need to have the Mandatory attribute set to protect the integrity of the records.

You can also set the Mandatory attribute for a field in a particular form. If you select the Mandatory attribute in the Structure editor, you cannot deselect it on a particular form. However, you can apply the Mandatory attribute on a form to a field that does *not* have this attribute in the Structure editor. For information about setting the Mandatory attribute for a field in a form, see the section "Setting the Enterable and Mandatory Attributes" on page 373.

Display Only

The user cannot enter values from the keyboard into a field that has the Display Only attribute set. You must use a default value for such a field or write a method that inserts a value in the field. A field with the Display Only attribute is useful for displaying values that you do not want database users to modify, such as calculated totals or a sequence number assigned by a method.

You can also make any field non-enterable on a particular form. For information about making a field non-enterable, see the section "Setting the Enterable and Mandatory Attributes" on page 373.

Can't Modify

If the Can't Modify attribute is set for a field, 4th Dimension accepts the value initially entered in the field, but does not allow the user to modify the value after the record has been saved. The user can edit an entry in such a field only during the initial creation of the record, before the record is accepted. Once the user saves the record, the value in the field not editable. The value can be modified by a method or by returning to the Design environment and removing the attribute.

Use Can't Modify for fields that must provide an audit trail such as Date Received, Date Paid, and so on. The Can't Modify attribute is often used for the field that uniquely identifies each record in the table.

Indexed

You should use the Indexed attribute for fields that you frequently use for searching and sorting. For example, you might index Last Name, Company name, or Product name if you plan to search for specific records or sort the records by these fields. You also use this attribute for fields that establish relations between tables. For more information, see "Setting Relation Properties" on page 200.

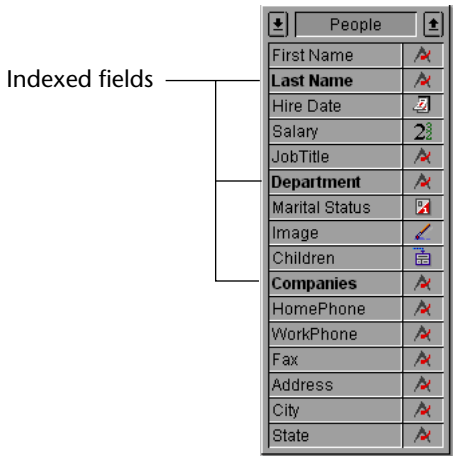
The Indexed attribute causes 4th Dimension to create an internal index table for the field. The table allows 4th Dimension to perform rapid searches and sorts on the field. When searching or sorting on an unindexed field, 4th Dimension moves through data sequentially, examining each record in order. An index allows 4th Dimension to search and sort without going through every record.

Note When selecting the Indexed attribute for a field in a table that has over 1,000 records, 4D allows you to choose between two indexing modes. For more information, refer to "Indexing or Reindexing a Field" on page 191.

You can index Alpha, Real, Integer, Long Integer, Time, Boolean, and Date fields. As you add and delete records, 4th Dimension automatically updates its index table. If you assign the Indexed attribute to an existing field, 4th Dimension automatically indexes the existing data when you leave the Design environment. You can specify as many indexed fields as you want.

Do not index every field. An index increases the size of the database, using more space on disk. Using many indexes also increases the time needed to save a record since 4th Dimension updates the index table with each entry.

Indexed fields are displayed in bold type in the Structure editor.



Unique

Use the Unique attribute when you want to be certain that each record has a different (unique) value in this field. The Unique attribute should be used for the field that uniquely identifies each record in the table. The Unique attribute is useful to validate fields that store Employee numbers, Social Security numbers, Purchase Order numbers, and so on.

If you want to set the Unique attribute for a field, you must first make it an indexed field. The Unique attribute is disabled unless the Indexed attribute is set.

The Unique attribute prevents duplication of empty values as well as actual entries. An empty field cannot be duplicated in another record.

Note If you apply the Unique attribute to a subfield (a field in a subtable), the attribute ensures that no subrecord contains a duplicate value. The Unique attribute applies to all subrecords, not just the subrecords for each parent record.

Invisible

You can make a field invisible in the User environment and custom applications by selecting the Invisible attribute for the field. Use the Invisible attribute if you inadvertently create an "extra" field or if a field is no longer needed in the current structure. The Invisible attribute hides the field from the user. A field with this attribute does not appear in all standard 4th Dimension editors and dialog boxes that appear in the User and Custom Menus environments.

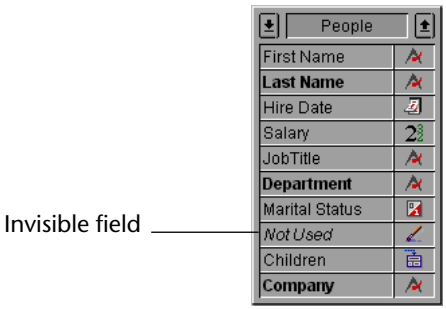
These editors and dialog boxes include the following:

- All query editors
- Order By editor
- Chart editor
- Label editor
- Quick Report editor
- Import and Export dialog boxes
- Apply Formula dialog box

In each of these places, the user is unable to see or choose the field. For instance, the user cannot choose an invisible field for a report created with the Quick Report editor.

Note When using the editors, users have the option of saving their specifications (e.g., the query or sort they created) to disk files. In this case, any invisible fields specified will be used in the operation. In addition, users can type the names of invisible fields in the Apply Formula dialog box.

Invisible fields are displayed in italics in the Structure editor window.



QuickTime Compression

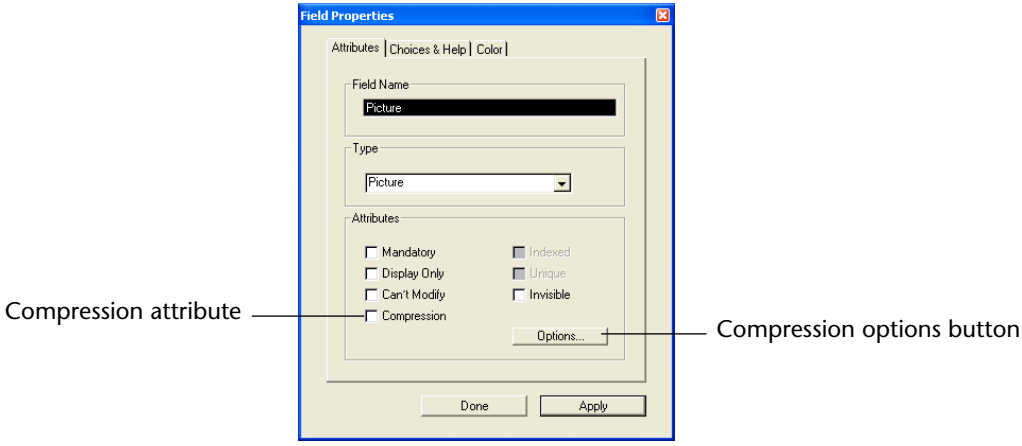
You can apply a QuickTime™ compression to picture fields. Selecting that attribute lowers the disk space required for storing pictures.

Under Windows, you must have QuickTime version 4 minimum installed on your machine.

The **Compression** attribute can be selected under the following conditions:

- The QuickTime™ extension is currently installed in the System folder. This extension is provided by Apple Computer, Inc.
- The field type is Picture.

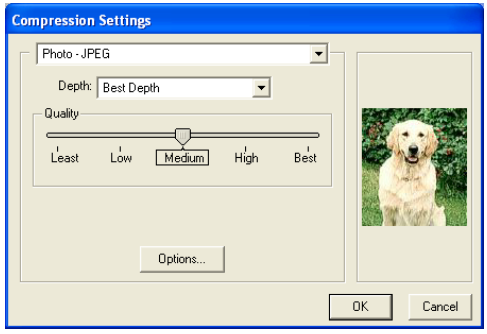
The **Compression** attribute is accessible in the **Attribute** page of the **Field Properties** palette.



If you check the **Compression** Attribute without setting any QuickTime™ compression options, the default compression is used. To remove picture compression, uncheck the **Compression** box.

Note QuickTime™ compression cannot be used with picture fields used to store plug-in areas.

The **Options** button allows you to set the QuickTime™ compression settings. The contents of the dialog box depends on the version of QuickTime™ that is installed on your machine. The following dialog box is from QuickTime 4.



Note The settings you define in the Compression Settings dialog box will apply to all the records of the table—for that field. However, the user is given the option to use different settings when pasting a picture into a field in the User or Custom Menus environment. To do so, press the **Option** (MacOS) or **Alt** (Windows) key while pasting the picture in the field to display the Compression Settings dialog box.

The QuickTime extension manages several compression methods, shown in the above illustration. For more information about these methods, refer to the QuickTime documentation available online from Apple Computer.

Note For more information about storing pictures, refer to the *4th Dimension Language Reference* Manual.

Choices and Help

The second page of the Field Properties window lets you assign a choice list to the field and enter a help message.

Choices

Use the Choices attribute if you want to display a choice list for entering information in the field. To use this attribute, you first need to create the choice list using the List editor.

Use the Choices attribute when you want to standardize entries in the field and avoid misspellings. Use a choice list for a field that has a limited number of valid entries or a limited number of usual entries. Using a choice list does not necessarily prevent the user from typing a different value (one that does not appear on the choice list). For more information about choice lists, see the chapter "Creating Lists," page 641.

You can also assign a choice list to a field on a particular form. However, when you assign a choice list only on a form, the list is not displayed in other editors and dialog boxes, such as the Query editor. For information about using a choice list in a form, see the section "Using Choice Lists," page 374.

Help

You can provide users with additional information about a field by adding a Tip or Balloon Help to the field. When the database is used on a Macintosh with Balloon Help turned on, the Balloon Help appears next to the field whenever a user places the pointer over the field in any form in which the field is included. A Tip is displayed on all platforms.

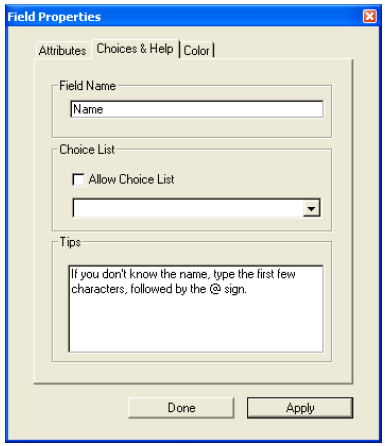
Note You can create Balloon Help when you are designing a database under Windows, but the Balloon Help can be displayed only under MacOS running System 7.0 or greater, with Balloon Help turned on.

► To add a help message to a field:

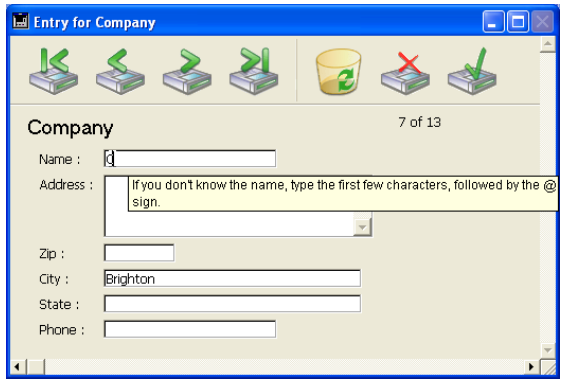
1 **Click the Choices & Help tab in the Field Properties window.**
The Choices & Help page appears.

2 **Enter the text in the Help message area and click Apply.**

The figure below shows text being entered into the message area. In this example, the help message is prompting the user about using the wildcard character during data entry.



When the user selects the field, the tip appears below the field, as shown below.



Setting the Color of the Field

The Color page of the Field Properties window lets you set the color of either the field name or the background of the row in the table image in which the field appears. You use a Design mode property in the Preferences dialog box to determine whether the color you choose applies to the name or to the background. For information on this, see the section “Structure Editor Page,” page 125.

► To set a color:

■ **In the Color page, click the desired color in the Color palette and hit the Apply button.**
OR
Right-click (Windows) the field name of the table or press the Ctrl key while clicking the field name of the table (MacOS); select Color from the contextual menu, then select the new color from the Color palette.
The field name or its background (depending on the Preference selected) is displayed in the color you have just chosen.

► To set the default color:

1 **Check the Default Color option.**
The default color of the field is re-applied.

Modifying Fields and Field Properties

You can change a field's name or properties at any time, whether or not you have entered data into the field.

► To modify a field's properties:

1 **Double-click the field name in the Structure editor.**
OR
Select the field you want to modify and choose Field Properties from the Structure menu (or in the contextual menu of the Structure window).
OR
Double-click the field name in the Tables page of the Explorer (or click the field name and click Edit).
4th Dimension displays the Field Properties window for that field. The window displays the field name and properties that you have previously assigned to the field.

2 **Make any necessary changes.**

3 **Click Apply to save the changes to the field properties.**

Renaming a Field

If you change the field name, 4th Dimension automatically updates the name anywhere the field is used (e.g, forms, methods, and disk files containing specifications from the 4th Dimension editors). Field labels on forms are not changed.

Field names are updated in methods as long as the methods are closed when the name is changed.

Changing a Field's Type

If you change the field type before entering any data into the field, 4th Dimension simply changes the field type. The only restriction is on the Subtable field type which cannot be changed to any other field type.

If you change the field type after entering data into the field, 4th Dimension converts the data to the new type if possible, when the data is loaded for the first time after the change. Data from a Picture field converted to any other type does not display. Data from a field converted to a Picture or Subtable field type does not display.

When you convert a field type, 4th Dimension retains the field's original value until you modify the record. For example, if a Text field contains text such as "over 10" and you change the field type to Integer, the modified field displays "10." If you change the field back to a Text field without editing the field value, 4th Dimension displays "over 10" again.

Changing a Field's Attributes

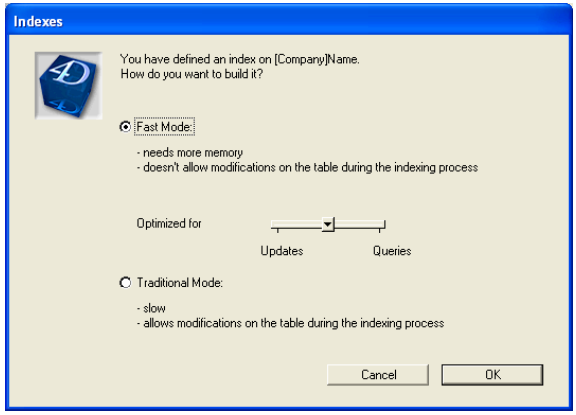
If you change field properties, the change does not usually affect data already entered into the field. For example, if you set the Unique attribute, only entries made after that are checked for uniqueness; field entries made up to that point may include duplicates. However, if you set the Indexed attribute, all field values (both old and new) are included in the index.

Indexing or Reindexing a Field

If you index a field (by selecting the **Indexed** attribute, by selecting the **Reindex** command of the Structure editor contextual menu or when defining a relation between two tables), both the new and old (created before the indexing) values will be indexed.

When indexing a field that has more than 1,000 records, 4th Dimension lets you choose between two indexing modes: the "traditional" mode, which is the mode used in previous versions of 4th Dimension, and the new "fast" mode, which in most cases is significantly faster.

The choice between the two modes is only given for tables that have over 1,000 records because below that value, the gain in speed is not significant. The mode selection dialog box is displayed each time you index or reindex a field.



To select a mode, click the corresponding button. If you select the Fast Mode, the optimization parameter must be set, depending on how the indexed field will be used. You use the slider to set the parameter to the desired value between two extremes:

- **Updates**: The index will be updated very frequently because data entry is being done.
- **Queries**: The index be updated rarely because the datafile is largely complete and the database is primarily used for searches, sorts, reports, etc.

The Optimization slider determines how the index is optimized, not its definition. Even if you place the slider to the extreme right for Queries, the index will still be modified when data is modified. However, its performance will not be optimal for data entry.

Which Mode to Choose? Each mode has its advantages and inconveniences:

Comparative table of the two index modes

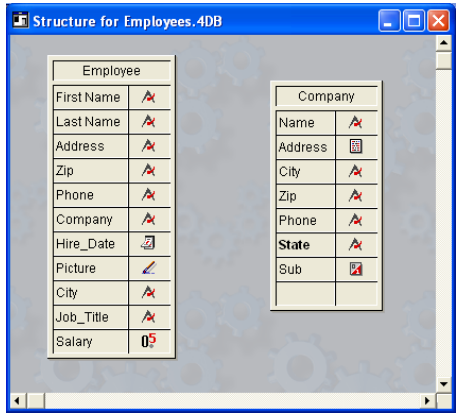
	Advantages	*Inconveniences*
Fast Mode	- Fast construction[1] of the index - Optimization possibility according to usage	- Needs more memory - "Locks" any modifications to the table during the indexing process
Traditional Mode	- Needs less memory - Allows you to modify the table during the indexing process	- Slow construction of the index - No optimization possible

1. The increase in speed depends mainly on the number of records to index, the fragmentation of the data, and the available memory (RAM).

To summarize, the fast mode (which is selected by default) is more efficient, can be optimized and should be used in most cases. To obtain the maximum increase in speed, indexing must be done as often as possible in memory (you can verify, in the Evaluation page of the Runtime Explorer, if the memory cache is sufficient to contain the index). However, if you don't have enough memory or if you don't want a locked table in your database (in write mode) for a certain amount of time, you can choose the traditional mode.

Relating Tables

You will usually need to create structures in which several tables share information. For instance, suppose you create a database to keep track of people and their companies. The database structure, shown below, contains a [People] table for storing people information and a [Companies] table for storing company information.



Although useful, the information stored in each separate table does not fulfill your information tracking needs. When you are viewing a record from the [People] table, you need to be able to view information about the company for which that person works and when you are viewing a record from the [Company] table, you need to be able to view information about all the people who work for that company.

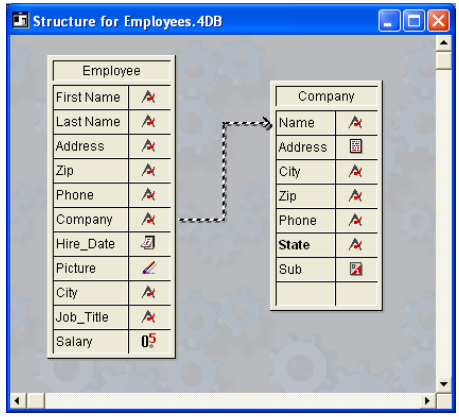
To allow two tables to share information in this way, the tables can be related to each other — in other words, a relation can be established between the data in each table.

In 4th Dimension, *table relations* allow data stored in one table to be accessed from another table. Tables that share information by means of a relation are called *related tables*.

Relating tables allows you to do the following:

- Store data efficiently,
- Update data in one place and have the change reflected everywhere the data is used,
- View related information,
- Perform queries and sorts in one table that are based on data in another table,
- Create, modify, or delete records in related tables.

The figure below shows a relation created between the [People] table and the [Companies] table in the Structure editor.



The [People] table contains one record per person. The [Companies] table contains one record per company. The relation between the two tables allows you to access, enter, modify, or delete information from both tables. For example,

- When a person's record is onscreen, you can view or modify the corresponding company information — the address, city, state, zip code, and company telephone number.
- When you add a new person, you can link the person's record to the appropriate company record (if the company is already entered), or, if the person's employer is not in the database, create the new company record while creating the person record. For more information, see the section "Entering Data in Related Tables," page 211.
- For each company, you can view or modify information for each person in the company — name, title, telephone number, and so forth. You can also add a person record from within the company record. For more information, see the section "One to Many Properties," page 203.

Related Fields

You are able to display information from related tables by means of the related fields — the fields that connect the two tables in a relation.

The basic purpose for relating tables is to instruct 4th Dimension which record or records to make current in one table based on which record is current in the other table. The related tables make use of data in two related fields to identify corresponding records. In the following example, the company name is stored in both the [People] table and the [Companies] table.

Related fields

First Name	Last Name	Title	Company
Biff	Davis	Salesperson	Howard Battery Co.
Andy	Venable	Engineer	Howard Battery Co.
Bryan	Pfaff	Secretary	Howard Battery Co.
Kathy	Forbes	Secretary	Howard Battery Co.

Company Name	City	St	Phone
Howard Battery Co.	Arcadia	CA	818-576-2534

The Company field in the [People] table and the Company Name field in the [Companies] table relate the two tables. The Company Name field in the [Companies] table is the *primary key field* for [Companies]. It uniquely identifies each company record. A primary key should have the Indexed and Unique attributes. The Company field in the People table is a *foreign key field*. Each value of the foreign key field matches exactly one value of the primary key field in the related table. A foreign key should have the Indexed attribute.

If both the primary and foreign key fields do not have the Indexed attribute when you create the relation, 4th Dimension assigns this attribute automatically.

Each value in a foreign key field is equal to one value of the primary key field in another table. In this example, a value of the foreign key field in [People] matches exactly one value of the primary key field in [Companies]. The foreign key field is also indexed but its values are non-unique (e.g., several people may work for the same company).

In some database designs, the values of the primary key field are assigned by the database automatically — either by assigning a sequence number that 4th Dimension generates or by a user-written method. Such a procedure guarantees the uniqueness of the key field. For example, if the primary key field in the [Companies] table is a sequence number rather than the company name, it would be possible for users to enter several companies with the same name but different addresses. Also, if a company name changes, the user could make the change to the database without disturbing the relation between the two tables.

If the user is permitted to enter the value of the primary key field, you should use both the Unique and Can't Modify attributes to check for uniqueness of the initial entry and to prevent users from subsequently changing the entry to a non-unique value. If you elect not to use the Can't Modify attribute, you will need to take other measures to prevent users from creating "orphaned" records in any related tables by making changes to the values of the primary key field.

When relations are established, you can read and write values in one table while working in the related table. For example, when you enter a company name in a person's record, 4th Dimension searches for that company in the [Companies] table and displays the company address and phone number in that person's record. When you view a company's record, 4th Dimension searches in the [People] table for all the people who work at that company and displays their records in the company record.

These relations can be invoked automatically (i.e., with no programming on your part) or you can choose to use manual relations. In the latter case, you use methods to load and unload related records and control the creation, modification, or deletion of related records. Manual relations are sometimes preferable in complicated structures in which more than two tables are related to one another and you need to control the loading and unloading of related records.

You can choose to use automatic relations by selecting the appropriate properties at the time the relation between the tables is specified. For more information, see the section "Automatic and Manual Relations," page 210.

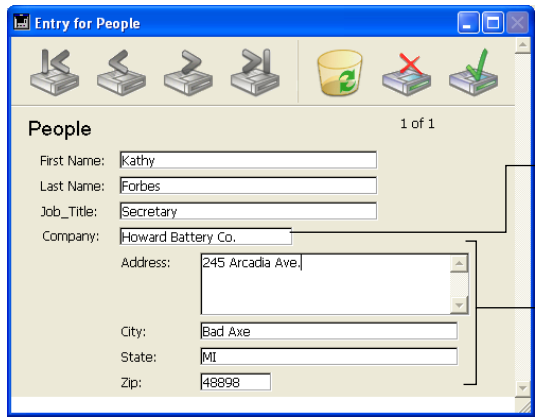
The One Table and the Many Table

When you create a relation between two tables, the table containing the primary key in the relation is called the *One table* and the table containing the foreign key in the relation is called the *Many table*. The tables are called the One table and the Many table because one record in the One table relates to many records in the Many table and many records in the Many table relate to one record in the One table. This type of table relation is called a *Many to One* relation.

In the relation between people and companies, the [Companies] table is the One table and the [People] table is the Many table. One company record relates to several people (i.e., all the people who work for that company) and several people relate to one company (i.e, the company for which they work). For instance, there may be one record for Acme in the [Companies] table but many records of people employed by Acme in the [People] table.

When any record in the [People] table is made current, 4th Dimension loads the corresponding single record from the [Companies] table. If any fields have been included from the [Companies] table, the values for these fields are automatically displayed. For information about including fields from other tables, see the section "Selecting Fields from Related One Tables," page 238.

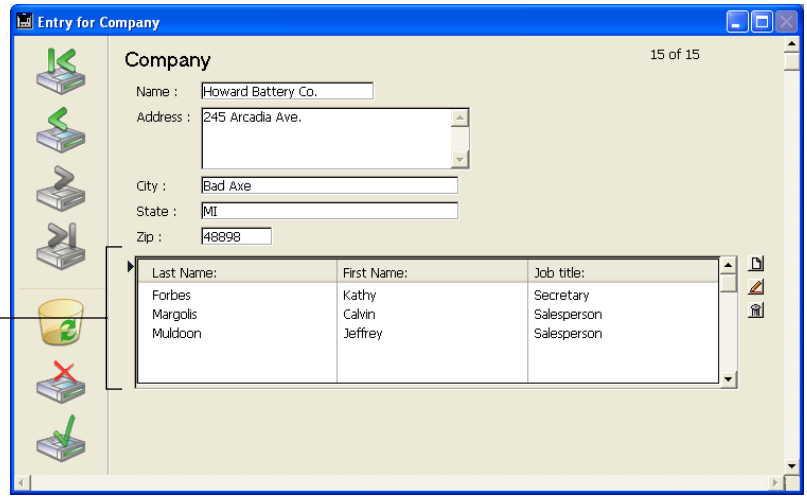
The figure below shows how the company name in a [People] table record specifies one record in the [Companies] table so that the [People] table record can display the company's address and phone number.



Conversely, when a record in the [Companies] table is made current, 4th Dimension creates a selection of records in the [People] table and displays them on the form. Since the relation specifies several records in the other table, the names and titles of many people can be displayed.

Note Only records currently displayed on the form are loaded into memory.

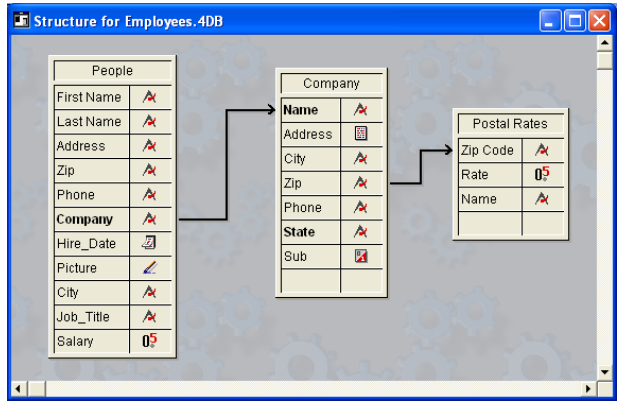
The figure below shows how a company name in a record in the [Companies] table specifies several records in the [People] table so that the [Companies] table record can display a list of people employed by that company.



The distinction between the One table and the Many table is specific to a particular relation. A table may be the One table in one relation and the Many table in another. A table in a relation need have only one primary key, but it can have several foreign keys.

For example, suppose you decide to send a package of sample merchandise to everyone in your [People] table. You add a [Postal Rates] table that contains zip codes and the postal rate for each zip code. Using this structure enables you to print an address label for each person that includes the amount of postage needed to mail the package.

The figure below shows the [Postal Rates] table added to the database structure.



The Zip Code field in the [Postal Rates] table is its primary key, so the [Postal Rates] table is the One table. The Zip field in the [Companies] table is the foreign key field for this relation. Since the Zip field is a foreign key, it can have non-unique values. The Zip field will contain duplicate Zip codes for companies that are near each other. The [Companies] table is therefore the Many table in relation to the [Postal Rates] table.

Whether a table is a One table or a Many table, therefore, depends on its relation to the other table. The [Companies] table is the Many table in relation to the [Postal Rates] table and it is the One table in relation to the [People] table.

Setting Relation Properties

You must have at least two tables in your database to create a relation. You create a relation by drawing a line between two fields.

The field where you start drawing must be a foreign key field in the Many table and the field where you end must be the primary key field in the One table. Using the Company database example in this chapter, you would start drawing in the [People] table and end in the [Companies] table. Remember,

You always draw a relation from the Many table to the One table.

The related fields must have the same field type. For example, the Company field in the [People] table and the Company Name field in the [Companies] table can be related because they are both Alpha fields.

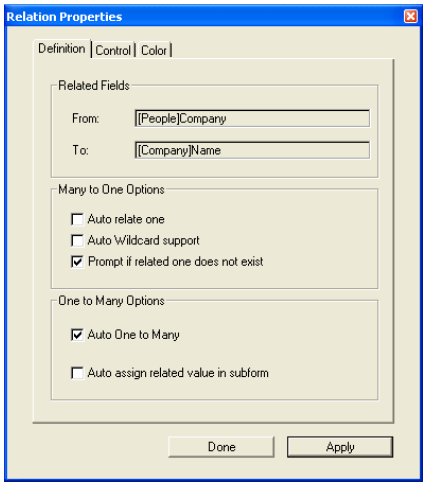
You can use these field types for the primary and foreign key fields:

- Alpha
- Number (Real, Integer, or Long Integer)
- Time
- Boolean
- Date

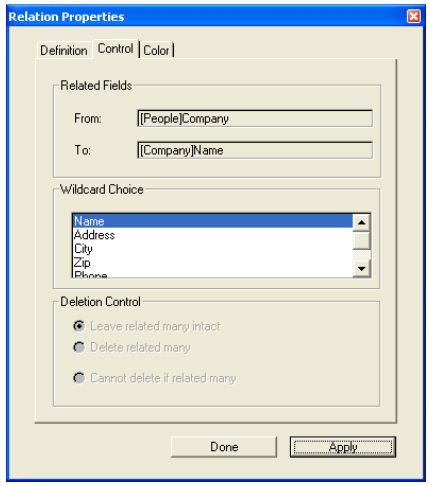
Primary and foreign key fields must be indexed. If the fields do not have the Indexed attribute set, 4th Dimension automatically indexes them when you switch to the User environment.

Note If you draw a relation between tables that have 1,000 records, you will be prompted to select an indexing mode. For more information, refer to the paragraph “Indexing or Reindexing a Field,” page 191.

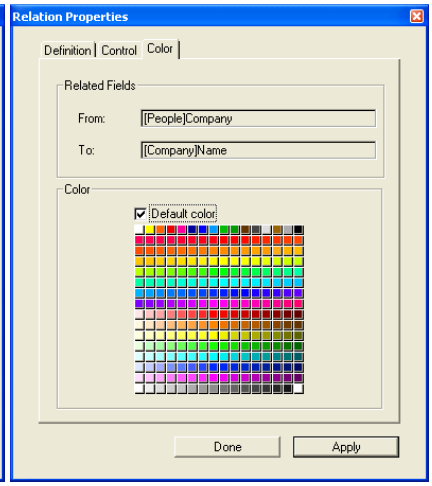
You specify various properties of the relation with the Relation Properties window. It has three pages: Definition, Control, and Color.



Specify the primary and foreign keys.
Set automatic relation options.



Specify the wildcard choice field in the One table.
Set Deletion Control options.



Specify the color of the relation line in the Structure editor.

Here are descriptions of the relation properties:

Related Fields

The Related Fields area identifies the foreign and primary key fields:

- The “From” field is the foreign key field in the Many table for this relation,
- The “To” field is the primary key field in the One table,

You draw the relation line *from* the foreign key field in the Many table *to* the primary key field in the One table.

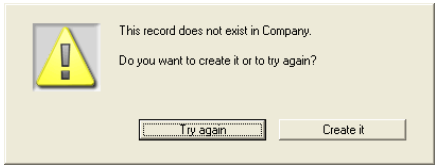
Many to One Properties

The Many to One properties affect what happens when a record from the Many table is opened:

- **Auto Relate One** This check box establishes automatic relations from the Many table to the One table. For example, when a record from the [People] table is opened in the User environment, the related company in the [Company] table is selected. This allows 4th Dimension to display information about the company for which the employee works if you so desire. If you deselect Auto Relate One, you can manage the loading and unloading of the related One record using commands in the language.

- **Auto Wildcard Support** This check box has the effect of invisibly appending the wildcard character (@) to any value entered in the foreign key field from the Many table when the user tabs or clicks out of the field. If the user enters a partial value, 4th Dimension looks for a matching value in the related One table. If 4th Dimension finds only one match, it automatically completes the entry. If 4th Dimension finds more than one possible match, the user is presented with a list of values from which to choose. For a complete description of this process, see the section "Entering Data in Related Tables," page 211.

- **Prompt if related One does not exist** This check box forces 4th Dimension to display a dialog box that lets a user create the related One record if it does not exist. By default, when you enter a value in a related field from the Many table, 4th Dimension checks to see if a matching record already exists in the related One table. If 4th Dimension cannot find a match, the following dialog box is displayed:



This dialog box allows the user to create a corresponding record in the One table while you are entering a record in the Many table. For instance, suppose that you have an Invoicing database that contains an [Invoices] table and a [Customers] table. If you enter an invoice in the [Invoices] table and the customer to whom the invoice belongs does not already have a record in the [Customers] table, 4th Dimension will ask you if you want to create the corresponding record in the [Customers] table when you validate the record in the [Invoices] table.

You can suppress this dialog box by unchecking the **Prompt if related One does not exist** check box. Suppressing this dialog box is useful when you need to manage the creation of the related One record using a method.

One to Many Properties

The One to Many properties control automatic relations in the other direction.

- The **Auto One to Many** check box establishes automatic relations from the One table to the Many table. For example, when a record from the [Company] table is opened in the User environment, the related records in the [People] table are loaded. This enables 4th Dimension to display the records of the people that work for the company in a subform.

  When you create a relation, the automatic relation check boxes are already selected. If you want to turn off automatic relations from the Many table to the One table, deselect the **Auto Relate One** check box. If you want to turn off automatic relations from the One table to the Many table, deselect the **Auto One to Many** check box.

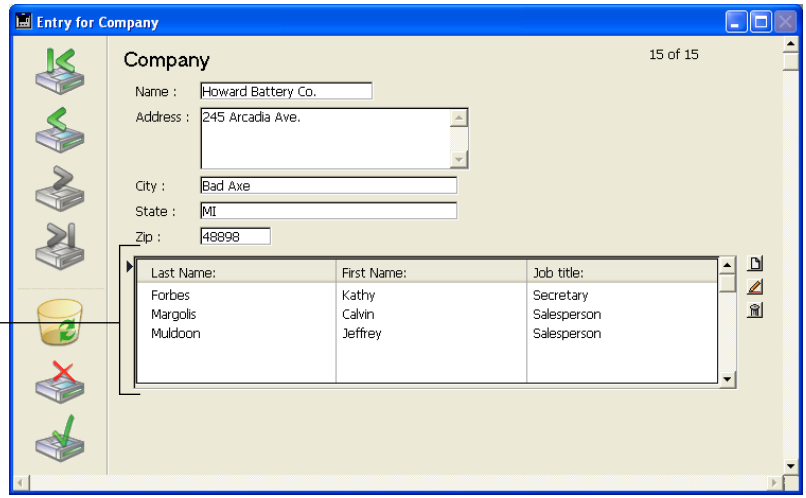
- **Auto Assign Related Value in Subform** is used to automatically assign the value of the primary key field in the One table to the foreign key field in the Many table during data entry. This option is available only if automatic One to Many relations are established.

  This option affects data entry when an input form in a One table has a subform of a related Many table (for information on subforms, see the section "Adding a Subform to the Form," page 255). If **Auto Assign Related Value in Subform** is selected, a user can add records to the subform (i.e., the related Many table) and have the relating value assigned automatically[1]. This occurs as long as the record is added by typing **Ctrl+/** (**Ctrl–Enter** under MacOS) or by pressing the **Add to Subform** button.

  In the relation between the [Companies] table and the [People] table, the [Companies] table is the One table and the [People] table is a related Many table. Each company has one record in the [Companies] table and several records in the [People] table.

1. This works when the foreign key field is *not* displayed in the subform.

The data entry screen for the [Companies] table is shown below.



If you want to add a person record from this screen, you need to assign the Company name to the foreign key field of the person's record. Otherwise, the new record would not be related to the correct record in the [Companies] table. **Auto Assign Related Value in Subform** does this automatically. If you deselect this option, you would need to make the assignment using a method.

Note If you double-click in a blank area of the subform and proceed directly to the input form to add a new record, or if you modify the relating field value in the One table after you have created records in the Many table, **Auto-assign Related Value in Subform** has no effect and you must either manually assign the relating field value or use the language.

Wildcard Choice

The Wildcard Choice list allows you to select an additional field to display in the Wildcard choice list (which appears when the user enters the 4th Dimension wildcard symbol (@) in the relating field during data entry). Usually, you will want to select the field that best identifies the record. For more information about the wildcard choice field, see the section "Using Wildcard Choice Lists," page 212.

Deletion Control

The Deletion control options regulate record deletion in the Many table when a record is deleted in the One table. Normally, the user cannot delete records in a table unless it is the current table. This means, for example, that to delete records from the [People] table, it must first be made the current table. You make a table the current table by choosing it in the Choose Table/Form dialog box in the User environment.

When tables are related, 4th Dimension allows you to specify one of three special cases for records deleted from the One table.

The following deletion control options can be set only if the Allow Deletion Control check box in the Preferences dialog box is checked. For more information, see the section "Data Control Page," page 134.

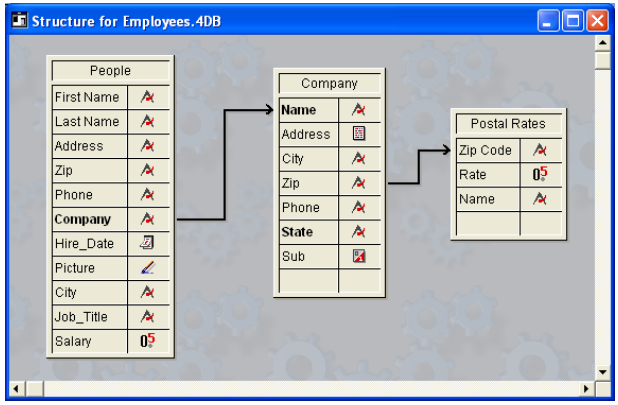
- **Leave Related Many Intact** Selecting this radio button allows the user to delete a record in the One table, leaving the corresponding records in the Many table intact. This leaves records in the Many table without any corresponding related record in the One table. The only effect is to render the information from the One table unavailable. No record from the One table is loaded when a record corresponding to the deleted record is loaded in the Many table.
- **Delete Related Many** Selecting this radio button instructs 4th Dimension to automatically delete all related records in the Many table when the user deletes a record in the One table. This property ensures that no related Many records become "orphaned" when the corresponding related One record is deleted.
- **Cannot Delete if Related Many** Selecting this radio button instructs 4th Dimension to prohibit the user from deleting a record in the One table if there are related records in the Many table. This property ensures that no records are mistakenly deleted.

  Notice that you can freely delete records from the Many table, no matter which choice is made.

  The **Delete Related Many** and **Cannot Delete if Related Many** radio buttons enforce what is called *referential integrity* in database theory. When referential integrity is in effect, 4th Dimension ensures that each record in a related Many table will always be associated with exactly one record in the related One table.

If you set the Deletion control option to either **Delete Related Many** or **Cannot Delete if Related Many**, 4th Dimension automatically adds the Indexed, Can't Modify, and Unique attributes to the primary key field in the One table. You cannot remove these attributes unless you first change the Deletion control setting to **Leave Related Many Intact**.

If you have several related tables, deletion control is activated for each relation as in a chain. For instance, suppose you have the structure shown below. If a Zip code is deleted from the [Postal Rates] table (a One table) and **Delete Related Many** has been selected for each relation, 4th Dimension first deletes the records for the corresponding companies in the [Companies] table and then deletes the records of all the people who work for those companies in the [People] table.



When confronted with contradictory Deletion control settings, 4th Dimension does not allow the deletion to occur. For instance, if **Delete Related Many** is selected for the relation between the [Companies] table and the [Postal Rates] table but **Cannot Delete if Related Many** is selected for the relation between the [People] table and the [Companies] table, no deletion will occur and the records in the [Companies] and [People] tables will remain intact.

The Deletion control choice is made when the relation is established. To change the choice, you can modify the relation's properties. For complete information about establishing and reestablishing relations, see the section "Creating a Relation Between Tables" on page 207.

Color

The Color property controls the color of the relation line in the Structure editor. The color of the relation line does not signify any other property of the relation.

Creating a Relation Between Tables

You create a relation by dragging from the foreign key field to the primary key field. You can do this using either the table images in the Structure editor window or the Explorer.

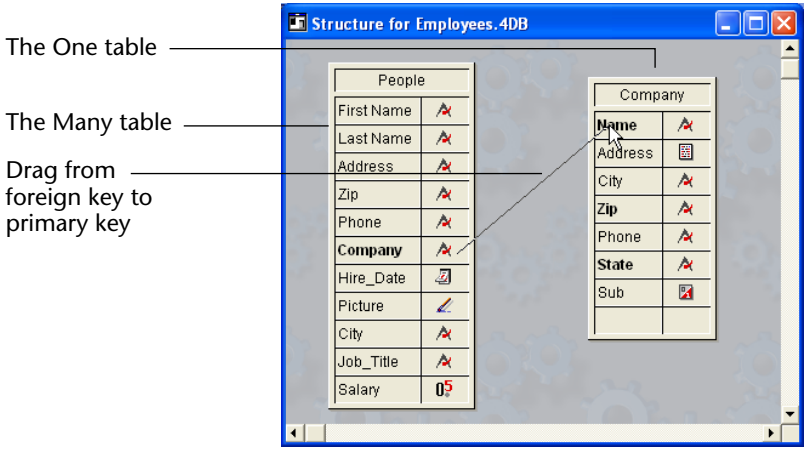
It is convenient to use the Explorer to create the relation when your structure is large and the table images of the tables that you want to relate are not adjacent to each other. Using the Explorer method, only one of the two table images must be visible in the Structure editor window.

Tip To center a table image on-screen in the Structure editor window, double-click its name in the Tables page of the Explorer.

► To create a relation using the Structure editor window:

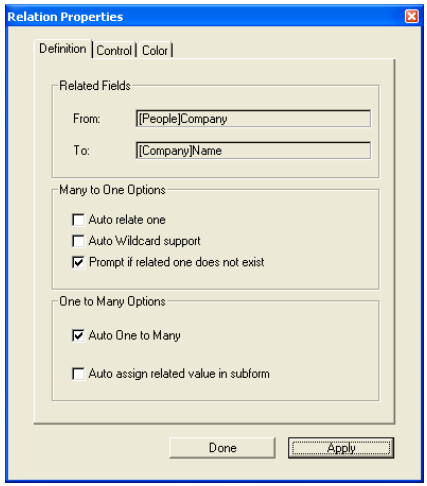
1 **In the Structure editor window, move the pointer over the foreign key field for this relation.**

2 **Hold down the mouse button and drag toward the table to be related.**
As you move the pointer, 4th Dimension selects the field and draws a thin line, as shown below.



3 **Drag to the primary key field in the One table and release the mouse button.**

The Definition page of the Relation Properties window appears.



For information on setting properties for the relation, see the section "Setting Relation Properties," page 200.

Configuring relations

This section describes the process of setting relation properties. The properties are the same regardless of the method you use to create the relation.

► To set relation properties:

1 **Check to make sure that the foreign and primary key fields are correct.**

Remember:

- The "From" field is the foreign key in the Many table for this relation,
- The "To" field is the primary key in the One table.
- The primary and foreign key fields must be of the same data type.

2 **Select the desired automatic relation check boxes to establish automatic relations between the tables.**
OR
Deselect the check boxes to establish manual relations.

In automatic relations, whenever a record from the One table is used, the related record or records in the other table is made the current selection for that table.

3 **Click the Control tab and use the Wildcard Choice list to select the additional field to display in the Selection window.**

For information on how the Wildcard choice field works during data entry, see the section “Wildcard Choice,” page 204.

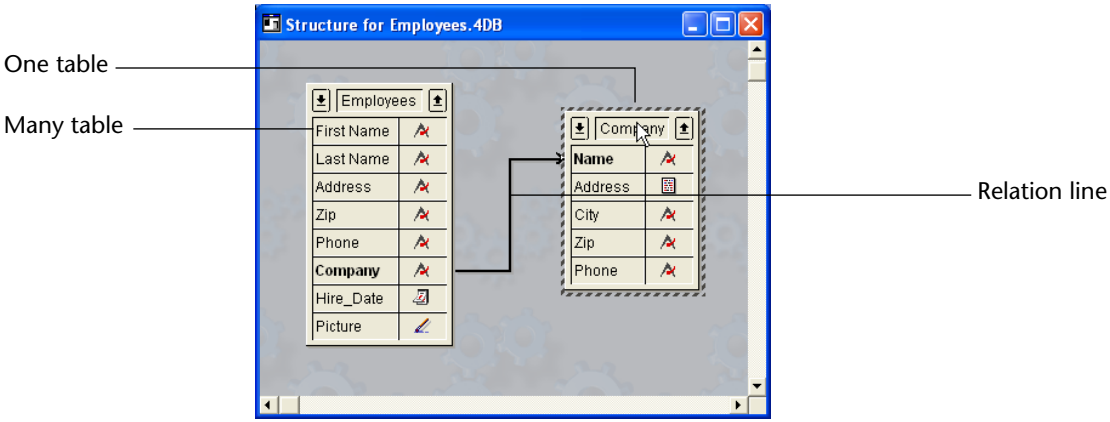
4 **Select a Deletion control option by clicking the radio button for the choice you want.**

The Deletion control radio buttons allow you to determine what happens when you delete a record in the One table. The **Leave Related Many Intact** radio button is the default. To be able to change this default option, you must have selected the **Allow Deletion Control** check box in the Preferences dialog box. Otherwise, the deletion control options are dimmed (refer to the section “Data Control Page,” page 134). For more information about deletion control options, see the section “Wildcard Choice,” page 204.

5 **Click the Color tab and set the color of the relation line (optional).**

6 **Click the Apply button.**

4th Dimension displays the Structure editor window with a line between the two fields, as shown in the following illustration.



The arrow on the relation line points to the One table.

Removing Relations

4th Dimension lets you remove a relation by deleting the line that relates the two fields.

► To remove a relation:

1 **In the Structure editor window, move the pointer over the name of the foreign key field in the Many table.**

2 **Click and drag over any empty area between table images.**

3 **Release the mouse button.**

4th Dimension removes the arrow and the tables are no longer related.

Reestablishing Relations

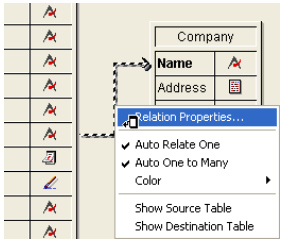
You can re-establish any relation at any time. You would do so, for example, if you mistakenly draw the relation between the wrong fields or if you want to change a relation property. 4th Dimension lets you reestablish a relation simply by drawing the relation line again.

To re-establish a relation using the same two fields, draw the relation line again. 4th Dimension displays the Relation Properties window so that you can make any necessary changes.

To re-establish a relation using a different field in the Many table, first remove the faulty relation and then draw the correct relation line again.

Modifying Relation Properties

If you need to view or change the properties of a relation, you can do so without having to recreate the relation.



► To modify a relation's properties:

1 **Move the pointer to the relation line until it changes to a relation icon .**

2 **Double-click the relation line.**
OR
Right-click (under Windows) or Control+click (under MacOS) on a relation, then choose Relation Properties... in the contextual menu.
The Relation Properties window appears, showing the selected relation's properties.

Note The Relate One and Relate Many properties as well as the color property can be selected using the contextual menu that appears when you right-click the relation line (Windows) or when you press the **Ctrl** key while right-clicking the relation (MacOS).

3 **Modify the relation properties as necessary and click Apply.**

Automatic and Manual Relations

Relations can be either automatic or manual.

In an automatic relation, whenever a record in a related table is made current, 4th Dimension selects the corresponding record or records. The record or records so specified can then be viewed, printed, modified, or used in searches and sorts. No programming is required.

In a manual relation, you dictate whether 4th Dimension loads the corresponding record or records into memory. To exercise this control, you use methods. For complete information about creating the methods that control related tables, see the *4th Dimension Language Reference* manual.

You would use a manual relation if you wanted to optimize the performance of specific applications that do not need all corresponding records loaded each time. For example, if your structure relates three or more tables together, you may want to control when related records are loaded into memory. You would also use a manual relation if you wanted to relate two tables with two separate relations. Only one automatic relation can exist between two tables. Any number of manual relations can exist between two tables.

Entering Data in Related Tables

You can display fields from the One table on a form for a related table. The user can use these fields to enter and edit the data directly while in the records of the related table.

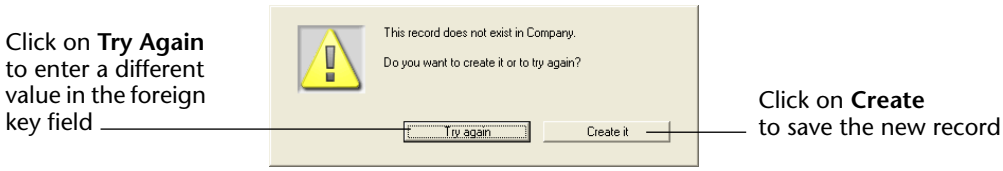
If the relation is automatic, information entered into the related fields is automatically saved in the related field's table. For complete information, see the section "Selecting Fields for the Form," page 237.

If the relation is manual, you use the language to display values in related fields and to save the information entered into related tables. You use methods to transfer and save any entered data.

The user enters records in the One table as in any other table — by typing information into an input form or by importing the data. For instance, you might have a subform in the [Companies] table that displays employee data from the [People] table. If automatic Relate Many relations have been established, any information you add or modify in the subform is automatically updated in the [People] table. For more information about subforms, see the section "Adding a Subform to the Form," page 255.

Occasionally, a user needs to create a new record for the One table while creating a record in the Many table. For example, suppose that, while creating a record in the [People] table the user enters a company name that doesn't exist in the [Companies] table. If automatic Relate One relations have been established, 4th Dimension automatically gives the user the opportunity to create a new record in the [Companies] table.

If the **Prompt if Related One does not Exist** property is selected, 4th Dimension displays the following message when a user enters a company name that does not exist in the [Companies] table.



4th Dimension requires that the corresponding record exist in the related table. The chance to create a new record in the One table is automatically provided to the user.

For complete details on entering data into fields from related tables, see the *4th Dimension User Reference*.

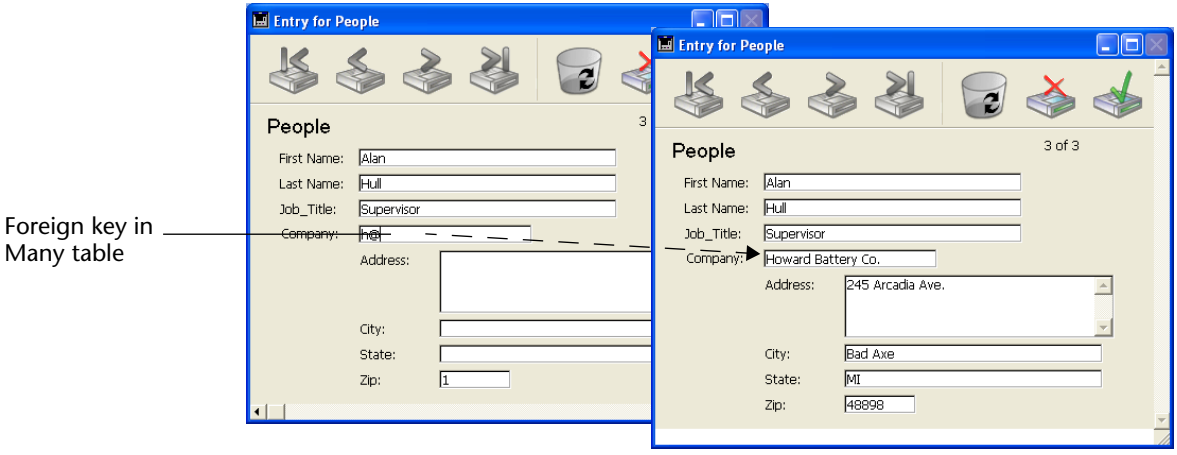
Using Wildcard Choice Lists

When tables are related, 4th Dimension allows the user to look up values in the One table when entering data into the foreign key field in the Many table. The user simply uses the standard wildcard character (@) in the related field. Doing so causes 4th Dimension to search for the corresponding entry in the related One table.

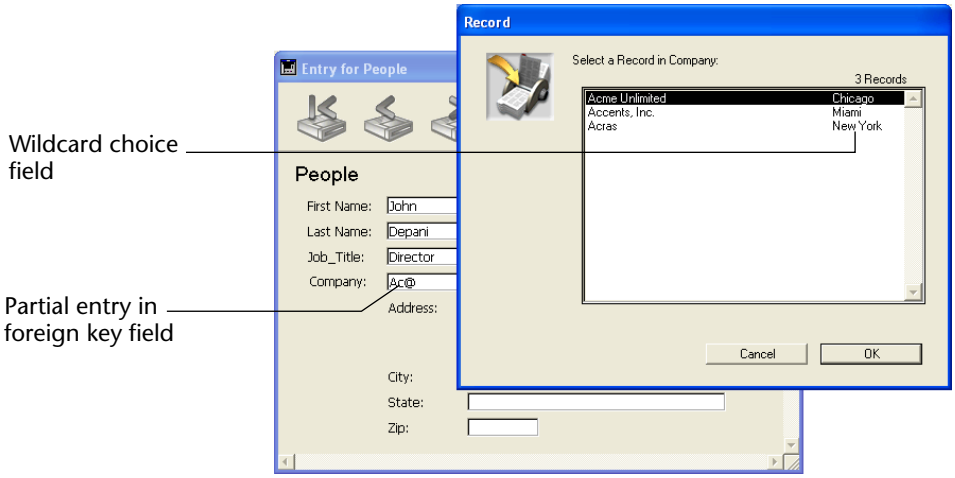
The wildcard character can be used in two ways: to complete a partial entry or to display a list of valid entries. When a list is displayed, the user can select the entry from the list. An additional field can be displayed with the related field.

For example, suppose the user is creating a record in the [People] table. Instead of typing 4D, Inc. in the Company field, the user can type Ac@ and then press Tab to move to the next field. Because @ is the 4th Dimension wildcard character, this entry means "this value starts with "Ac" and is followed by anything else." 4th Dimension looks in the related table for the record which matches this entry. If it finds one, it completes the entry and selects the next field in the data entry order.

The figure below shows how this use of the wildcard works.



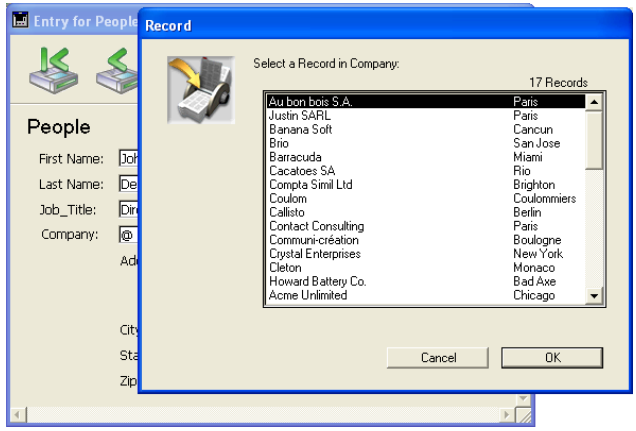
If 4th Dimension finds more than one entry that meets the requirement, it displays a list of entries so that the user can select the correct one. The figure below shows such a list being displayed.



You can specify a second field for the list to help the user decide which company to select. The second field is the wildcard choice field you selected in the Relation Properties window when you created the relation.

The figure above shows the list of companies displaying the city as well as the company name. This wildcard choice field assists the user who doesn't know whether the company is named 4D, Inc. or Acme Unlimited, but remembers that the company is located in Cupertino.

To see a list of all companies in the [Companies] table, the user enters @ only. 4th Dimension then displays a list of all the companies so that the user can select the correct one. The figure below shows a complete list of companies being displayed.



Note The record selection window can be resized.

Relation Types

Until now, we have been discussing the process of creating the most common type of relation — a relation between a One table and a Many table — called a Many to One relation. However, you can also create One to One and Many to Many relations. These types of relations are described in this section.

One to One Relations

One to One relations are used only in special cases since tables that are related on a one-to-one basis could be combined into a single table. Here are some reasons to use a one-to-one relation:

- You have large BLOB fields in the database. BLOBs would slow down the database if they were loaded into memory when a record is made current. By placing the BLOBs in another table, you can load the BLOBs only when needed.
- You have a very large number of fields and need to divide them into logical groups. Separate tables can make the database faster and easier to use.

- You want to limit access to certain fields. If you use separate tables, you can assign different access privileges to each table.

Many to Many Relations

Sometimes you need to relate many records in one table to many records in another table. This is called a Many to Many relation.

An example of a Many to Many relation is a database that tracks class enrollment. Suppose that this database has two tables, [Students] and [Classes]. A student may enroll in many classes and a class may have many students. You want to see all the classes that a student has enrolled in and you want to see all the students enrolled in each class.

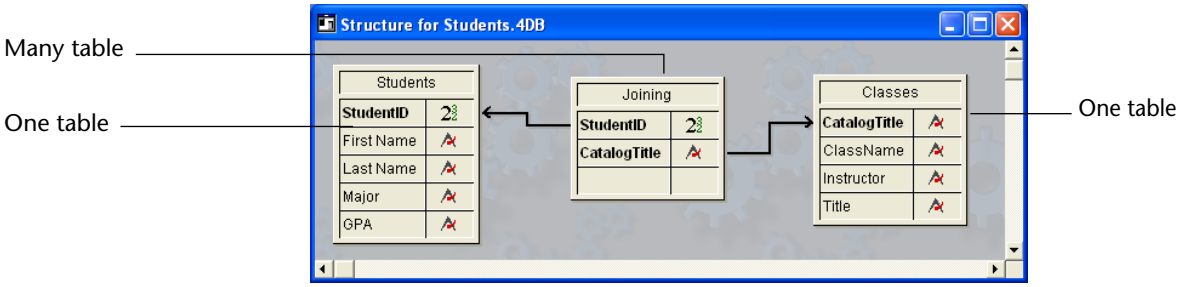
Other examples of Many to Many relations include the following.

- **[Suppliers] and [Products]** Each supplier provides many products and each product may be provided by several suppliers.
- **[Employees] and [Account]** Each employee works on many accounts and each account may be worked on by several employees.
- **[Movies] and [Actors]** Each movie involves several actors and each actor may appear in several movies.

You can use 4th Dimension to create automatic Many to Many relations. The key is to create an intermediate table which is related to the other tables using Many to One relations. You can then create input and output forms that handle all the necessary record tracking and data display.

This section describes how to use automatic relations to handle a Many to Many relation.

The figure below shows the enrollment database with three tables, [Students], [Classes], and [Joining]. This database structure is used throughout this section to explain how an automatic Many to Many relation works.



The [Students] table is a One table. It contains one record for each student, including their name, major, and GPA. The Student ID field identifies each student uniquely.

The [Classes] table is also a One table. It contains one record for each class, including the class name and the instructor. The CatalogTitle field identifies each class uniquely.

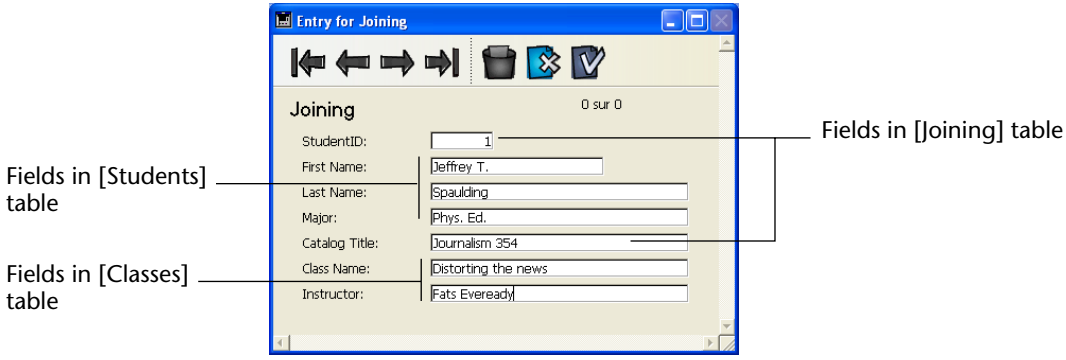
An intermediate table, [Joining], is the Many table for both of the other tables. It contains records for many students and many classes. Forms for this table are used for entering data into both of the other tables, and for displaying information in each of the other tables.

The use of three tables ensures that the data is stored efficiently. A student's complete record is stored only once. Each class has one record, stored only once. Records that relate students to classes are stored once for each enrollment. All of the information, however, is available in any combination.

Entering Data with Many to Many Relations

You use the intermediate table — in this example the [Joining] table — to enter and display information from both of the other tables. Each record that you enter in the [Joining] table is related to both of the other tables (a student and a class). The records from the [Joining] table contain only the two pieces of information that establish the relation: the student ID and the catalog title.

Here is an example of a new record being entered in the [Joining] table.

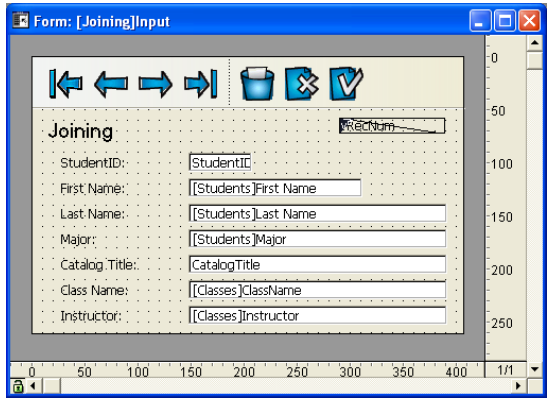


This record defines Jeffrey T. Spaulding as enrolled in a Journalism class. This record actually combines information from the other two tables.

A similar record exists for each class in which the student is enrolled. Only the Student ID and Catalog Title fields are actually stored in the [Joining] table. Each record catalogs a particular student taking a particular class.

Note When a record in the [Joining] table is loaded (as in creating such a record), it automatically creates a selection of records in the related tables. The selection consists of the corresponding student and class records. If you switch to either of the other tables, only a single record is displayed. To display all the records, choose **Show All** from the **Queries** menu.

The input form for this record is shown below. Notice that it contains fields from both the [Students] and [Classes] tables.



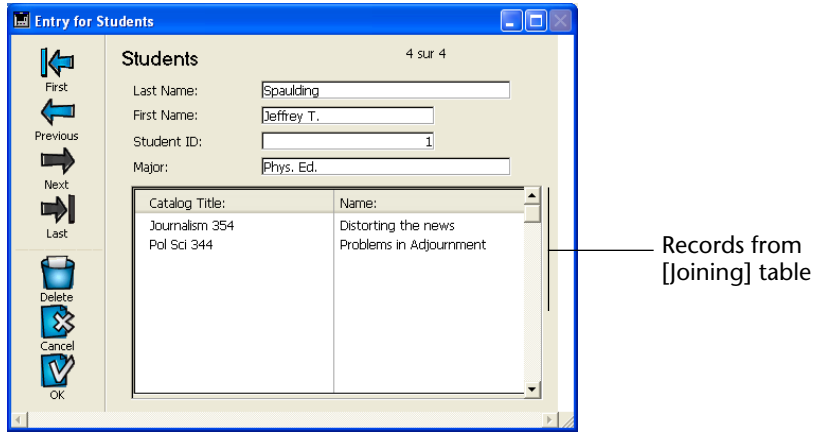
Data is entered only in the Student ID and Catalog Title fields. When a student ID is entered, 4th Dimension finds the student information in the related Students table and displays it in the Last Name, First Name, and Major fields. Likewise, when a Catalog Title is entered, 4th Dimension finds class information in the [Classes] table and displays it on the input form.

Displaying Information in a Subform

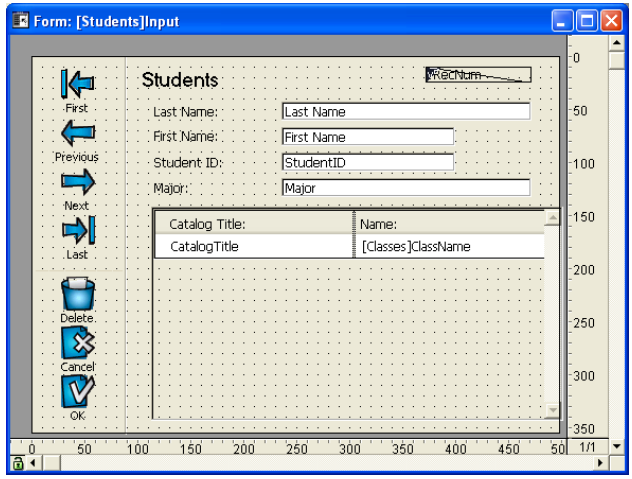
You can display information from these three tables using subforms. You can display all the classes a student is enrolled in on the student's record. You can display all the students enrolled in a particular class on the class record.

To display classes on a student's record, you use a subform. For information about creating subforms with the Form Wizard, see the section "Adding a Subform to the Form," page 468.

For information on creating a subform using the Form editor, see "Adding a Subform to the Form" on page 468.

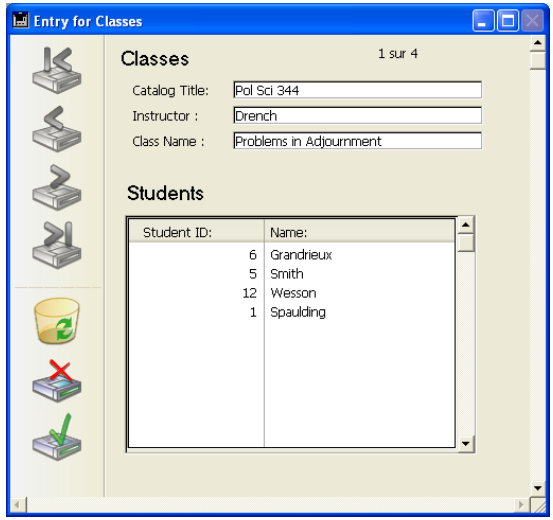


The record shown above is in the [Students] table. It shows information about the student at the top of the record. The information about the two classes that he is enrolled in is drawn from the [Joining] table where the enrollment information is kept.
Here is the form for this record:



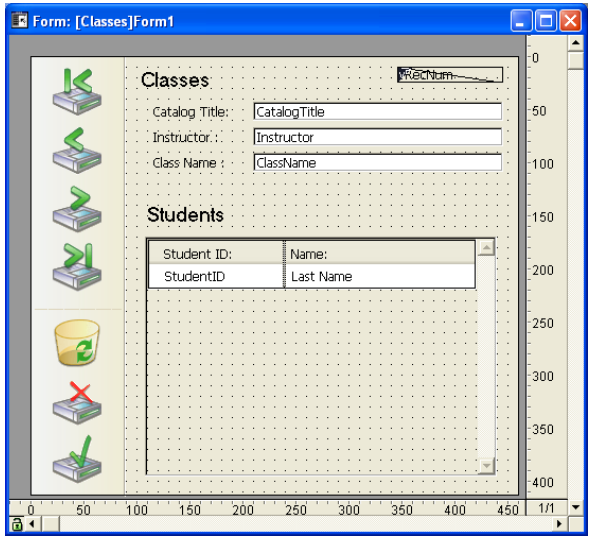
Notice that the subform is for the [Joining] table, not the [Classes] table. The [Joining] table contains the records that relate the student's record to the class records. The subform contains the ClassName field from the [Classes] table. Because of the relation between the [Joining] and [Classes] tables, 4th Dimension can display the correct class name automatically.

Here is a record that shows the students who are enrolled in a class:



This is a record from the [Classes] table. It shows class information and lists the students enrolled in the class. The information about the students is also drawn from the [Joining] table since that table contains the records that relate the classes to the students enrolled in them.

Here is the form for the record shown above:



In the above examples of subforms, you can enter records in any of the fields shown.

For example, to enter a new student into a class record, you simply tab to the last student record shown in the subform and press **Ctrl+/** (Windows) or **Command–Tab** (MacOS) to create a new record. When you enter the appropriate catalog title, the remainder of the information is entered in the record. For information about entering and deleting records in subforms, refer to the *4th Dimension User Reference*.

Creating Reports

Quick reports that include information from both the [Classes] and [Students] tables are typically generated from the intermediate table. If you create form reports for printing, you can use subforms to print information from either of the related One tables.

Here is a simple quick report that lists all the classes and the students that are enrolled in each class.

Class Name	First Name	Last Name
Auto Mech 101	Rufus T.	Firefly
Ceramics 211	Otis B.	Driftwood
	Jeffrey T.	Spaulding
Journalism 354	Jeffrey T.	Spaulding
Pol Sci 344	Jeffrey T.	Spaulding
	Otis B.	Driftwood
	Rufus T.	Firefly

For complete information about creating and printing quick reports, refer to the *4th Dimension User Reference*. For complete information about using forms for printing reports, refer to the chapter "Output Displays and Reports," page 477.

Analyzing Database Relations

The relations that you establish in a database play an important role in the operation of the database by controlling the flow of information between the tables.

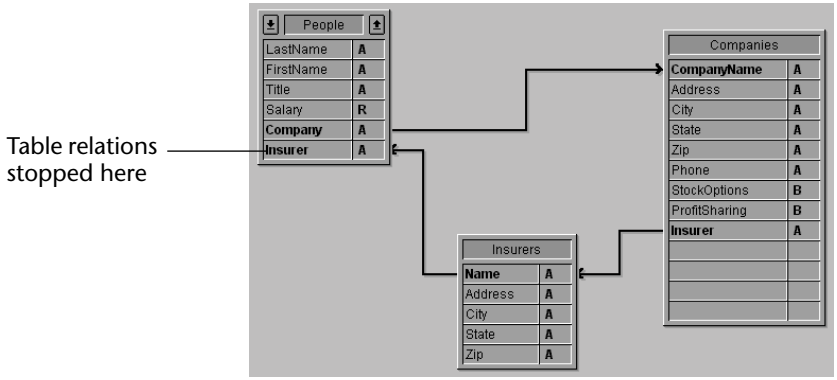
If a record with an automatic relation is loaded from disk using an input form, the corresponding record or records from the related table are selected. If a relation selects only one record in a related table, that record is loaded from disk. If a relation selects more than one record in a related table, a new current selection of records is created for that table and the first record in the current selection is loaded from disk. The record that is loaded from disk is called the *current record* for the table.

In the examples in this chapter, relations have been established between no more than three tables. In the real world, relations are often created between several tables and are activated one after the other, as in a chain. Each time a relation is activated, 4th Dimension creates a selection of records in the related table and loads a record from disk. The record that is loaded from disk becomes the current record for the table and — if the table has an automatic relation — 4th Dimension creates a selection and loads a current record in the next related table in the chain, and so on.

If the table relations have not been set up properly, the circulation of information between tables can become disorderly or corrupt. The following cases alert you to relational structures of which you should be aware.

Circular Relations

A circular relation is one in which table relations are set up so that the transfer of information will loop indefinitely. The figure below shows a circular relation in which the [People] table relates to the [Company] table, which relates to the [Insurer] table, which relates back to the [People] table.



When a record in the [People] table is loaded from disk, 4th Dimension loads the related company record from the [Companies] table. This becomes the current record for the [Companies] table, which in turn loads the related insurer record from the [Insurer] table.

If the table relations were allowed to continue, the records related to this insurer (all the people insured by the company) would be selected in the [People] table and the first record in that selection would be the current record. Note that this current record may be different from the current record that started this progression. In this situation, 4th Dimension has no way of knowing which record is really the current record.

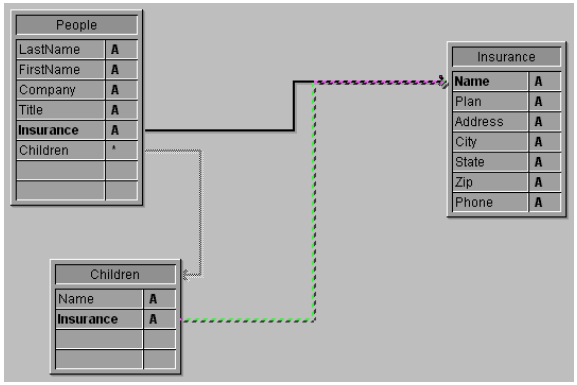
When 4th Dimension encounters this kind of circular relation, table relations are stopped at the last table in the chain. In this case, the relation between the [Insurers] table and the [People] table is not carried out.

Multiple Links to the Same Table

A similar conflict between current records occurs if you have more than one link to the same table.

Since you cannot have more than one current record at a time, you cannot manage an automatic table relation in which two or more tables are related to the same table.

The following illustration shows a database structure in which a table and its subtable both relate to the same table.



When a user is working with a record in the [People] table, the related record is loaded in the [Insurance] table and it is made the current record for that table.

However, there is also a relation between the [Children] subtable and the [Insurance] table. This means that another related record is loaded in the [Insurance] table based on the current record (the first record) in the [Children] subtable. If the child's insurance company is different from the parent's, this relational structure will cause problems.

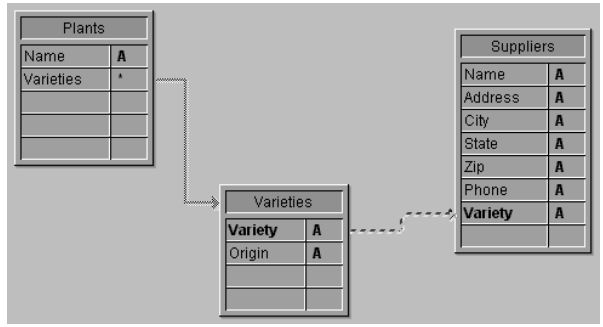
In this case, 4th Dimension does not stop the relations from proceeding. Both the relations are carried out, but not at the same time.

If you want to use this kind of structure, you must use manual table relations and control the relations using the commands described in the *4th Dimension Language Reference* manual.

Another example of a relational structure that cannot be managed by automatic relations is a structure in which one table has more than one relation to another table. Each time a user modifies either of the related fields in one table, the current record in the other table may change. In this situation, you cannot tell which relation is being activated.

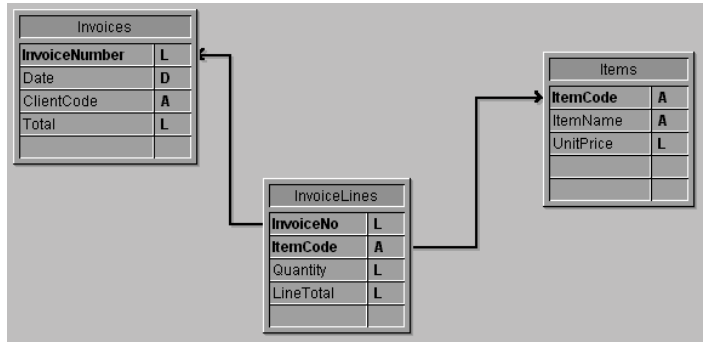
Relations from Multiple Records

Since there is only one current record in a table, relations are not established for all of the records in a selection. For instance, in the following figure, the records in the [Varieties] subtable are related to the records in the [Suppliers] table.



This structure implies that several varieties of plants are related to several suppliers and that the supplier records are loaded for all varieties of a plant species. However, the related supplier records are loaded for only the first record (i.e., the current record) in the subtable.

A similar case occurs in the Invoices database shown in the structure below.



When a record in the [Invoices] table is being used, a selection of records is created in the [InvoiceLines] table that contains all of the lines for that invoice. But the corresponding record in the [Items] table is selected only for the first item in the [InvoiceLines] table. The selection in the [Items] table does not include information about all the items in the invoice, only the first item.

However, if you place [InvoiceLines] in a subform in the [Invoices] table, *4th Dimension* calls each invoice line, one at a time, and activates the relationship for each one of them.

4 Creating Forms

Forms provide the interface through which information is entered, modified, and printed. A user interacts with the data in a database using forms and prints reports using forms.

Each table in your database generally has at least two forms. One form is for listing records on-screen and the other form displays one record at a time and is used for data entry and modification. The form that lists records is called the *output form* or *list form* and the form that displays one record at a time is called the *input form* or *detail form*. When you are viewing records using the list form, you can double-click a record to view the record using the current detail form.

This chapter covers the following topics:

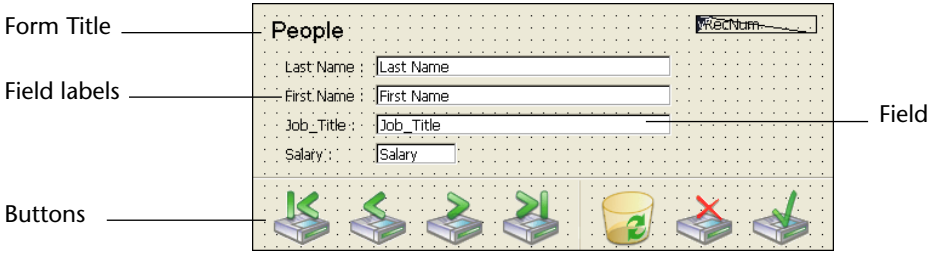
- Creating forms for data entry and display,
- Establishing default input and output forms,
- Deleting forms,
- Renaming forms.

For information about customizing forms, see chapter 5, "Form Editor Basics," page 267 and chapter 6, "Working with Fields and Active Objects," page 365. For information on creating and customizing list forms for listing records on-screen and for printing see chapter 7, "Output Displays and Reports," page 477.

About Forms

The form is the interface object that you use for data entry, for listing records, for printing reports and mailing labels[1], and (in custom applications) for custom dialog boxes and palettes.

4th Dimension lets you create standard forms quickly. It also provides powerful tools that let you create forms that implement sophisticated interfaces. Your forms can provide exactly what your database needs. With only point and click operations, you can create a basic form as shown below.



4th Dimension has two tools for creating and modifying forms, the Form Wizard and the Form editor.

The Form Wizard

The Form Wizard is your starting place for creating any type of form. With the Form Wizard, you can create a new form by choosing the desired fields from a list and the desired form template from a drop-down list. Form templates control the appearance of forms. A template specifies such characteristics as form size, platform interface, font attributes, and buttons.

1. You can also print reports and labels with the User environment's Quick Reports and Labels editors. These editors can also be added to a custom application.

The Form Editor

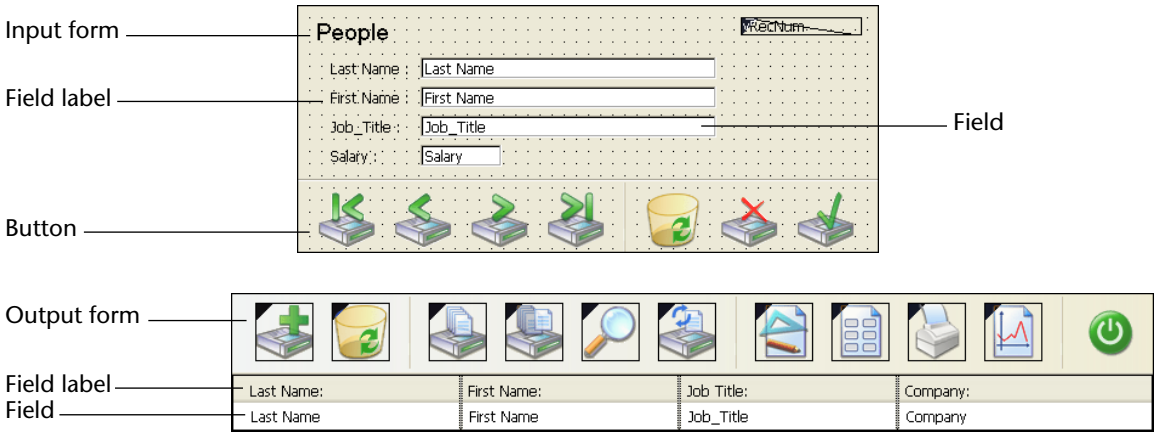
The Form editor is an object-oriented drawing environment that lets you customize forms by manipulating objects on the form directly. For example, you can reposition objects, add objects not supported by the Form Wizard, create multi-page forms with tab controls, enforce business rules by specifying data entry constraints, specify form access privileges, associate a custom menu bar with a form, and write form and object methods that run automatically when the form is used.

This chapter discusses creating forms for data entry and display using the Form Wizard. Chapters 5 and 6 discuss the Form editor.

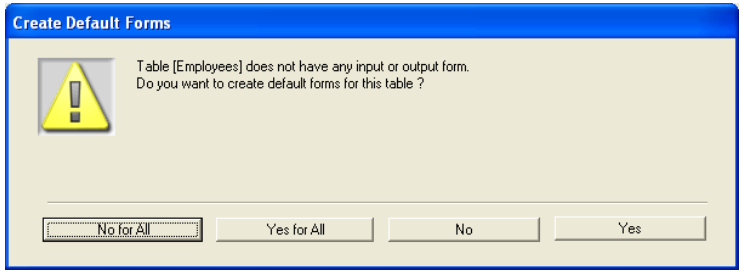
Forms, Tables, and Fields

Every form is attached to a table. The table to which a form is attached is called its *master table*. Each table must have at least one form so that information can be entered into fields and displayed on screen. Typically, a table has separate input and output forms. The input form is the one used for data entry. It displays one record per screen and typically has buttons for saving and canceling modifications to the record and for navigating from record to record (i.e., First Record, Last Record, Previous Record, Next Record). The output form displays a list of records, with one line per record. The results of queries are shown in the output form and the user can double-click a row in an output form to display the input form for that record.

The following illustrations show a typical input and output form:



If you switch to the User or Custom Menus environment before creating a form for a table, 4th Dimension asks you if you want to it to create default input and output forms for you.



Note With the Automatic Form Creation option in the Preferences, you can, for example, set 4th Dimension to automatically create default forms and therefore not display the Create Default Form dialog box. For more information, refer to the paragraph "Options Page," page 127.

Click **Yes** to create default forms. You can always return to the Design environment and modify them or replace them with more sophisticated forms. Without making any modifications, you can start using these forms to enter and display data in your database.

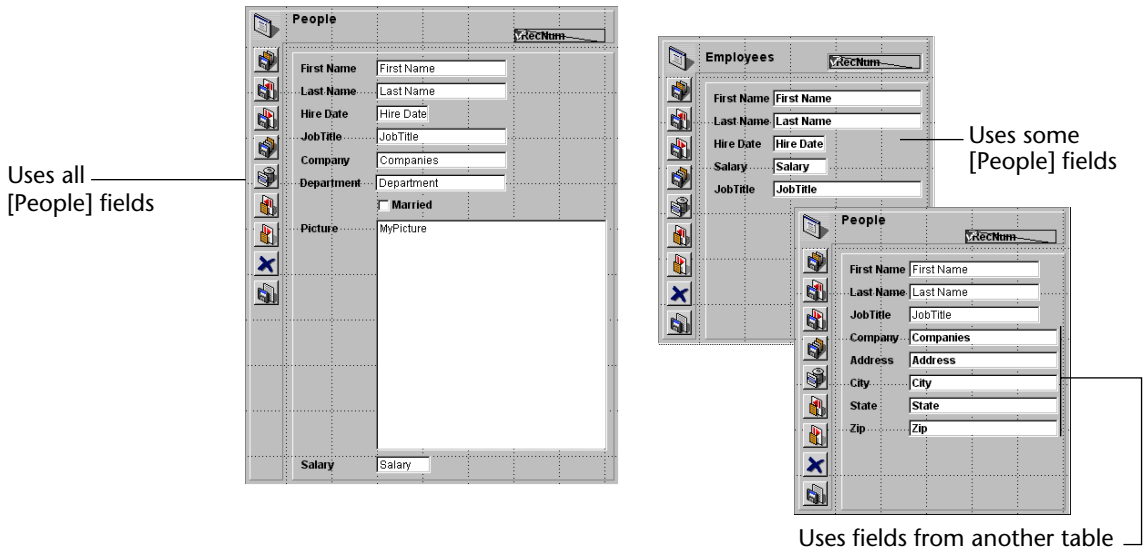
Your database can use a large number of forms that perform specific functions. In custom applications you can use the language to control which forms are the active input and output forms. For example, you may want to switch sets of forms depending on whether the user is using a monochrome or color monitor. You can also use the language to use different sets of forms for Web browsers and 4D Client users. When you write a custom application you can create forms for use as custom dialog boxes or floating palettes. In custom applications, you can also use multiple processes to allow users to work with several forms simultaneously.

A form can display fields from more than one table. You can place fields from a related One table on a form and allow users to enter values directly into the related One table. You can also include a *subform* that displays a list of records from a related Many table. A subform displays a list of records from another table or a subtable in the master table. With a subform, the user can view, enter, and modify records in another table. This is sometimes known as a *master-detail relationship*. For example, an invoicing application would use a subform on the invoicing input form that lets the user enter line items for the invoice.

Although the line items appear on the invoicing screen, the line item records are actually stored in a related Many table.

A form used for data entry can have more than one subform. For example, a contacts manager database can use a subform for telephone numbers, another subform for ToDo's, and another subform for prior contacts with the person. Each subform displays records from a different related Many table.

A particular form can use some of the fields in a table or all of the fields. You might have two input forms, for example — one for use by a clerk and one for use by supervisors — neither of which contains all the fields. You might use another group of fields for the screen display and yet a fourth group for a printed report.



Forms can be modified at any time, whether or not you have entered data into the database. Changes to a form do not affect the data stored on disk in any way.

Each form has one or more display pages in which fields and other enterable objects appear. If your fields don't fit on one page, you can create additional pages. When you create a multi-page form, you also add buttons or a tab control to allow users to move from one page to another.

Each form also has a background page (a page zero) on which you place objects that appear on all display pages. Use the background page to place background graphics, buttons, a tab control, and other graphic objects that define the "look" of the page, such as rectangles and labels.

Note When a multi-page form is used as an output form (e.g., when it is printed), only the first display page appears.

Active Objects and Graphic Objects

There are two kinds of objects in a form: active objects and graphic objects.

Active objects perform operations on data or provide a customized user interface. Active objects include the following:

- Fields, including fields from other tables
- Enterable and non-enterable areas (variables) for entering or displaying data
- Buttons (standard, 3D, Highlight, Invisible, or Picture) that perform actions
- Tab controls
- Combo boxes
- Pop-up menus and drop-down lists
- Hierarchical pop-up menus and hierarchical lists
- Radio buttons and check boxes used for entering values into Boolean fields or variables
- Thermometers, rulers, or dials that show relative values
- Lists and hierarchical lists, that allow the user to select from the list or drag elements to or from the list
- Graph and Plug-ins areas
- Splitters for resizing form areas
- Subforms that display data from other tables and subtables.

Graphic Objects

Graphic objects are geometric or textual elements that enhance the appearance of the form. Graphic objects include the following:

- Rectangles, ovals, and circles for enhancing the appearance of a form
- Text, for labelling areas in the form
- Grids, for aligning other objects
- Graphics from applications other than 4th Dimension, including pictures stored in the Picture Library

All objects, whether active or graphic, are handled in the same way in the Form editor — they are created by being drawn or dragged and dropped; they can be selected and moved or resized; they can be duplicated, cut, copied, and pasted; they can be aligned to each other or to an invisible grid; and their appearance can be changed.

Object Properties

Each object has a set of properties. For graphic objects, properties include foreground and background color, line width and fill pattern, resizing and repositioning options, and font size and attributes. For active objects, properties may also include the object's relationship to data, the object's "action" when it is used, a help message, and the object's method. When the Form Wizard generates a form, it assigns appropriate default properties to both graphic and active objects[1]; you can modify these properties in the Form editor.

Graphic objects have no impact on the data. You can create a graphic object on a form simply by drawing it and making any necessary modifications to its appearance. For example, when you create a text area, you draw the area, then you type the text you want to display. You can change the text at any time without affecting the data.

Active objects require instructions about their relation to the data or instructions on the actions that they are to perform. In simple cases, The Form Wizard does everything for you. Entry areas for fields are automatically associated with the appropriate field in the database structure and buttons automatically perform the appropriate actions. In other cases, you can specify additional instructions in the Form editor using the Object Properties window.

---

1. For example, the Form Wizard generates buttons that perform standard actions and assigns appropriate resizing and repositioning options to decorative rectangles.

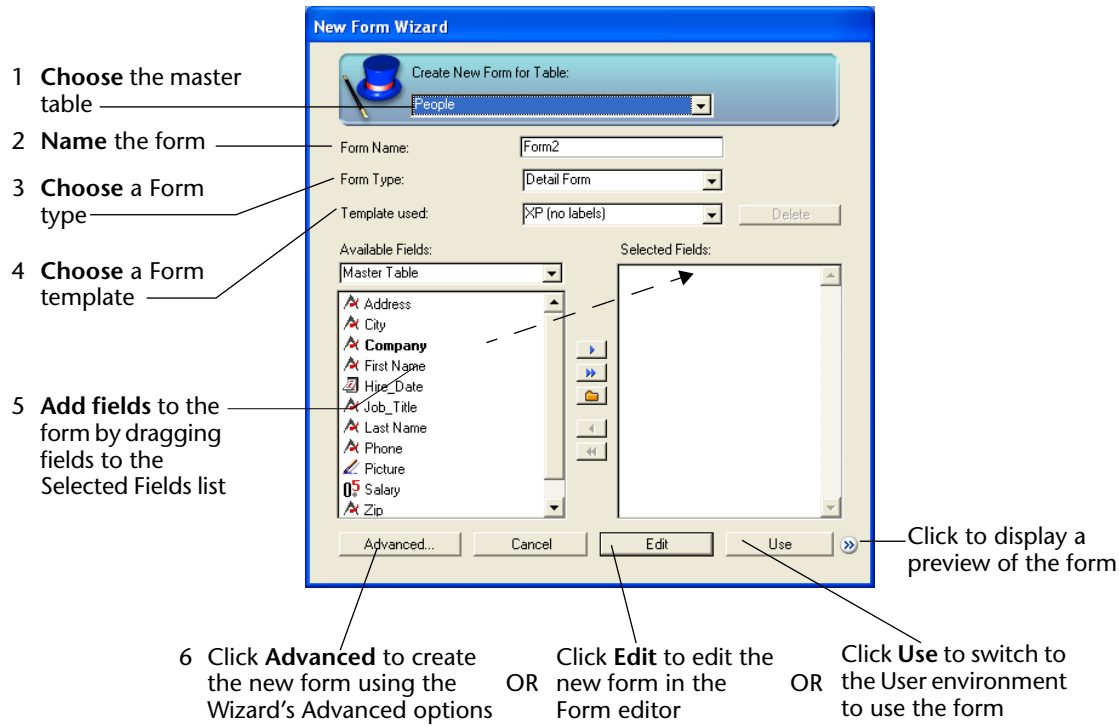
For example, you can assign special instructions to a button by writing a method. The method remains attached to the object as one of its properties. If the object is copied and pasted, it retains all its properties, including its method.

Chapter 5 provides detailed information about working with the Form editor. Chapter 6 provides complete information about active objects.

The Form Wizard

You can create new forms quickly with the Form Wizard. You can use a new form immediately after creating it or choose to edit the form using the Form editor.

The Form Wizard has two screens. The Basic screen lets you create new forms with a few simple operations. The steps for creating a form are shown in the following illustration.



If you want to create a standard form quickly, use the Basic screen.

Here are the basic operations:

- Name the new form by filling in the Form Name area.
- Choose a form type from the Form Type drop-down list. Your choices are:
  - **Detail Form** an input form for entering and modifying individual records,
  - **List Form** an output form for listing records on the screen,
  - **Detail Form For Printing** a form for printing individual records with one page per record,
  - **List Form for Printing** a form for printing a list of records, with several records per page.
- Choose a template from the Template drop-down list. A template controls many aspects of the appearance of the form. The Template drop-down list includes a variety of templates that ship with 4th Dimension and any user-defined templates that you create with the Advanced options in the Form Wizard. For information on adding templates to this list, see the section "Creating a Form Template" on page 258.
- Choose the fields for the form by dragging the desired fields in the Fields area to the Selected fields area (to its right). You can also use the arrows found in the middle of the window.
  As you add fields or change the form type or template, your changes are reflected in the preview area on the right.

  If you need more control over the appearance of the new form, you have two choices:

- **Use the Advanced options in the Form Wizard** Click **Advanced**... to customize the new form with the Form Wizard. The Advanced options let you set the font attributes of fields and labels, the platform interface for the form, appearance of fields and field labels, form size, the form background, associate a menu, choose custom buttons, and add a subform.
- **Use the Form editor** Click **Edit** to continue designing the form using the Form editor.

  If you are finished creating the form, you can click **Use** to switch to the User environment to use the form.

  For complete information on using the Form Wizard, see the section "Creating a New Form" on page 235.

The Form Editor

The Form editor is a powerful object-oriented drawing environment. Each of your forms can be displayed in a separate window and several forms can be open the same time. Objects on the form can be created or manipulated with the tools in the Form editor's Tools palette. You can set each form's properties and each form object's properties. While you are designing a form in the Form editor, you can switch to the User environment to test the form without having to first close the Form editor.

The Form editor offers many customization options that are not available in the Form Wizard, including the ability to:

- Set access privileges for the form,
- Assign properties to each object,
- Attach a method to the form and to each form object,
- Add pictures from the Picture Library to the form,
- Resize and reposition each object directly,
- Add types of interface objects that are not supported by the Forms Wizard, including tab controls, drop-down lists and pop-up menus, picture buttons, combo boxes, check boxes and radio buttons, hierarchical menus and hierarchical lists, scrollable areas, plug-ins, graphs, splitters, and additional subforms.[1]
- Set drag and drop properties for individual objects,
- Set data entry controls such as minimum, maximum, and default values, entry filters, and choice lists,
- Customize automatic resizing and repositioning options for each object.

For a complete description of the Form editor, see Chapters 5 and 6.

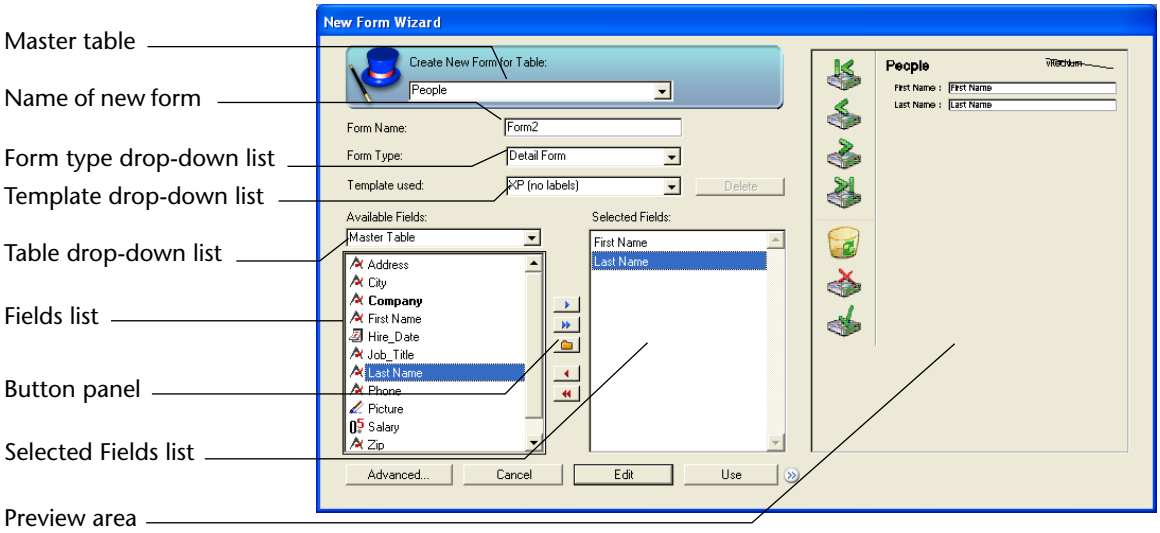
1. The Advanced options in the Form Wizard lets you add only one subform.

Creating a New Form

This section gives the basic steps for creating a form using the Basic screen of the Form Wizard. For information on the Advanced options of the Forms Wizard, see "Using the Form Wizard's Advanced Options" on page 243.

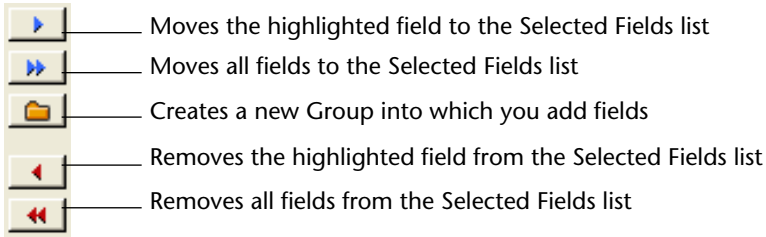
► To create a new form:

1 **Choose New Form from the Design menu.**
OR
Highlight the table name in the Forms page of the Explorer and click New.
4th Dimension displays the Basic screen of the Form Wizard.



The name of the master table is shown at the top of the window. Its fields are listed by alphabetical order in the Fields list.

The Button panel contains shortcut tools for moving fields to and from the Selected Fields list.



2 **If you need to change the master table, choose the master table from the New Form for Table drop-down list.**

3 **Name the form by filling in a name in the Form Name area.**

You can refer to the form by name using the language.

4 **Choose a Form Type from the Form Type drop-down list.**

Your choices are:

- **Detail form** a form for data entry and modification.
- **List form** a form for listing records on the screen.
- **Detail form for Printing**[1] a printed report with one page per record, such as an invoice.
- **List form for Printing**[1] a printed report that list records.

5 **Choose a template for the form.**

The template controls several aspects of the appearance of the form, including font attributes, field label placement, the design of decorative rectangles surrounding fields, and platform interface.
4th Dimension ships with several templates and you can add use the Form Wizard to add custom templates to this list. For more information about adding custom templates, see the section "Creating a Form Template," page 258.

6 **Select the fields you want on your form.**

For complete information about selecting fields for the form, see the following section, "Selecting Fields for the Form" on page 237.

7 **If you want to edit the new form in the Form editor, click Edit.**
OR
If you want to switch to the User environment to use the form, click Use.
OR
If you want to customize the new form with the Forms Wizard's advanced options, click Advanced.

For information about the Advanced options, see the section "Using the Form Wizard's Advanced Options" on page 243.

1. For information on creating forms for printing, see Chapter 7.

Selecting Fields for the Form

You select the fields that you want to appear in the form in the Form Wizard. You can also use the Form editor to add fields to the form after it has been created.

You can select any fields of any type, except Blob fields.

You can create forms that include fields from:

- The master table
- A related One table
- Any table

The Subform page in the Advanced options screen lets you create subforms that display fields from:

- Subtables of the master table
- A related Many table
- Unrelated tables

The following sections explain how to select fields from the master table and from a related One table.

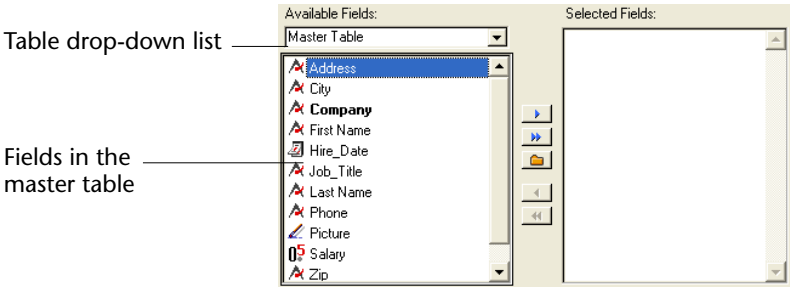
Selecting Fields from the Master Table

When **Master Table** is selected in the Table drop-down list, the Fields list displays a list of fields in the master table. Indexed fields are shown in bold.

▶ To select fields from the master table:

1 **If it is not already selected, choose Master table from the Table drop-down list.**

The master table is the table to which the form belongs.



2 Drag a field from the Fields list to the Selected Fields list.
OR
Click the field in the Fields list and click the Append button  .
OR
Double-click the field in the field list.
4th Dimension moves the selected field to the Selected Fields list adds the field to the preview area (if you have expanded the Form Wizard view).

Note If you want to include all fields on the form, click the **Add All** button  .

After you have selected fields for the form, you can reorder fields by dragging them vertically within the Selected Fields list.

3 Repeat this process for each field you want to add to the form.
To remove a field, select it and click the **Remove Field** button  . To remove all fields, click the **Remove All** button  .

If you are adding fields by dragging, you can insert a field in the list by dragging to the desired position in the Selected Fields list. Otherwise, 4th Dimension appends each field you add to the end of the list.

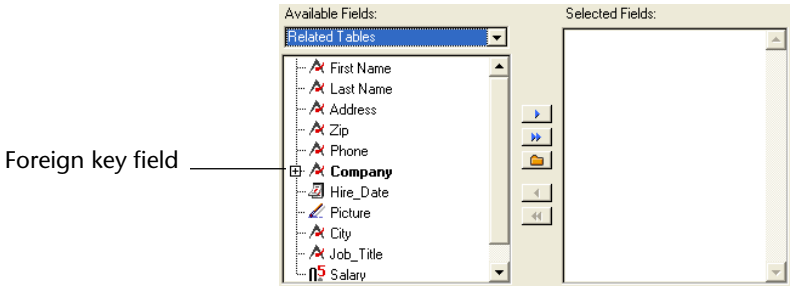
Note You can reorder fields in the Selected Fields list by dragging a field up or down.

Selecting Fields from Related One Tables

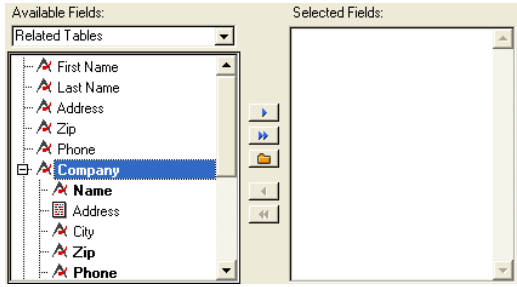
You can select fields from related One tables. 4th Dimension allows you to enter values directly into related One fields.

► To add fields from a related One table:

1 Choose Related Tables from the Tables drop-down list.
The Fields list changes to display a hierarchical list of fields in the master table. The foreign key fields are shown in bold and have a plus sign (Windows) or an arrow (MacOS).



2 **Expand a foreign key field to display the related One fields in the related table.**



3 **Add related One fields to the form using any of the methods used for adding fields from the Master table.**

As you add related One fields, they are shown in the preview area.

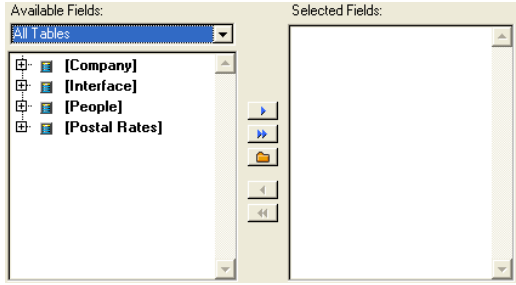
Selecting Fields From Other Tables

You can select fields from any table in the database. However, if the table is not the master table or an automatic related One table, you will need to use the language to manage data entry and display in the fields you select.

► To add fields from any table:

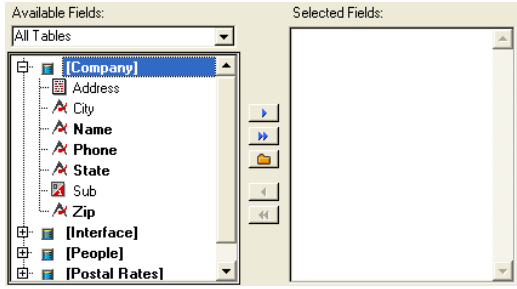
1 **Choose All Tables from the Tables drop-down list.**

A hierarchical list of all tables in the database appears in the Fields list.



2 **Expand the desired table to view its fields.**

The fields belonging to the table appear.



3 Add the desired fields to the form using any of the methods described in the previous section.

As you add fields, they appear in the preview area.

Reordering Fields

After you have added fields to the form, you can reorder them by dragging fields up or down in the Selected Fields list. Reordering a field list affects its position on the form. When you reorder fields, the preview area shows the effects of your changes.

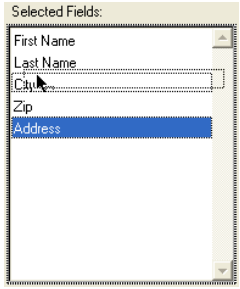
Note With the Form editor, you can rearrange fields by manipulating them directly.

► To reorder a field:

1 Select the field to be reordered in the Selected Fields list.

2 Drag the field up or down.

As you drag, the outline of the field indicates its new position. The following illustration shows the Address field being moved between the LastName and State fields.

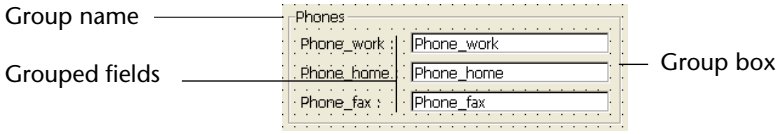


3 Drop the field at the desired location.

When you drop the field, the Selected Fields list changes to show the new field order and the preview area reflects your changes.

Grouping Fields

When you are creating a Detail form, you can define a group of fields. A group has its own label and set of fields in the group. A group looks like this:



You can create several groups in each form.

Note With the Form editor, you can also create group objects and move fields or other objects into or out of groups.

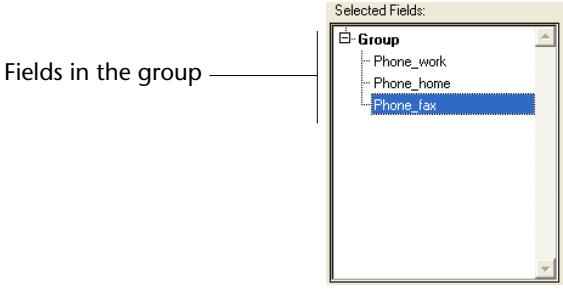
► To create a group:

1 **Click the Group button** .

A new group is created in the Selected Fields list. Its default name is "Group" and it appears as a hierarchical list.

2 **Add fields to the group by dragging the fields to the group name.**

As each field is added to the group, it appears below the group name.



When you are finished adding fields to the group, you can:

- Create another group,
- Add more ungrouped fields,
- Edit the form, use the form, or use the Advanced options.

- **To create another group, click the Group button and repeat the process of adding fields to the group.**
- **To add ungrouped fields, collapse the Group item in the Selected Fields list and continue adding fields.**
- **To edit the form, use the form, or use Advanced options, click the desired button at the bottom of the Form Wizard screen.**

Renaming the Group

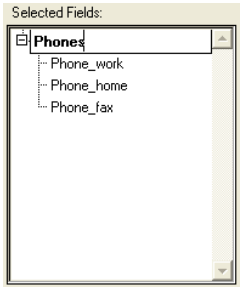
Unless you rename the group, the name of the group on the form will be the default name, "Group."

► To rename the group:

1 **Hold down the Command key (MacOS) or Ctrl key (Windows) and click the group name in the Selected Fields area.**

The name becomes editable.

2 **Type the new group name and click anywhere outside the text entry area to save the new name.**



When you save the new name, the preview area reflects your changes.

Reordering Fields in the Group

You can re-order fields in the group in the same way as ungrouped fields. Select a field to be reordered and drag up or down as described earlier in the section "Reordering Fields" on page 240.

Moving Fields from the Group

You can also move a field from a group to make it an ungrouped field or move the field to a different group.

- **To make a field an ungrouped field, drag the field from the group and move it diagonally in the direction of other ungrouped fields — to the above-left or below-left.**
- **To move a field to another group, make sure the target group is expanded, then drag the field to the field list in the other group.**

  If the destination group is empty, you first need to remove the field from the Selected Field and then insert it in the group as described above. 4th Dimension updates the preview area to reflect your changes.

Removing Fields

To remove a field from a form, highlight the field in the Selected Fields list and click the **Remove** button  . To remove all fields from the form, click the **Remove All** button  .

Using the Form Wizard's Advanced Options

The Form Wizard's Advanced screen also lets you create new forms with point-and-click operations but offers a wider variety of customization options.

The customization options depend on the form type that you select in the first screen of the Form Wizard. The Form Wizard supports the following form types:

- Detail forms
- List forms
- Detail forms for printing
- List forms for printing

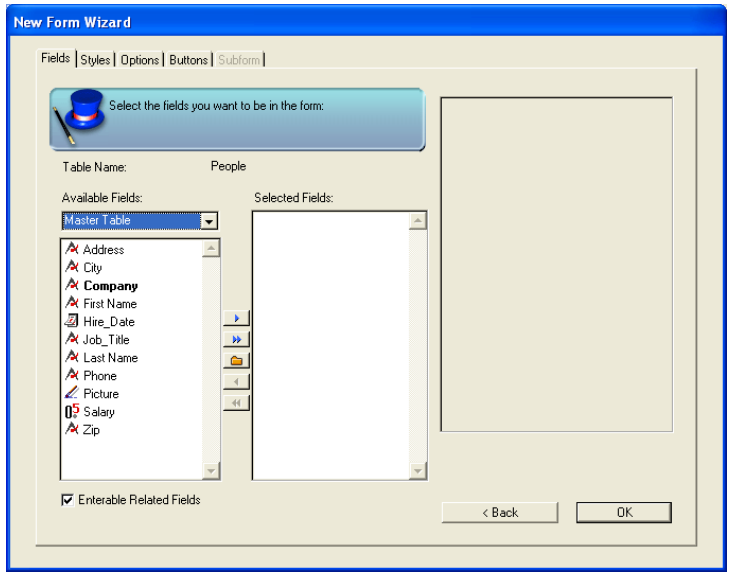
In addition, the Form Wizard lets you save your customization options as a template. The template name is added to the Template drop-down list that appears on the Basic screen of the Forms Wizard. With user-defined templates, you can quickly create highly customized forms from the Basic screen of the Form Wizard simply by selecting the desired fields and your custom template.

If you want to go back to the Basic Screen of the Form Wizard, you can do so by clicking the **<Back** button.

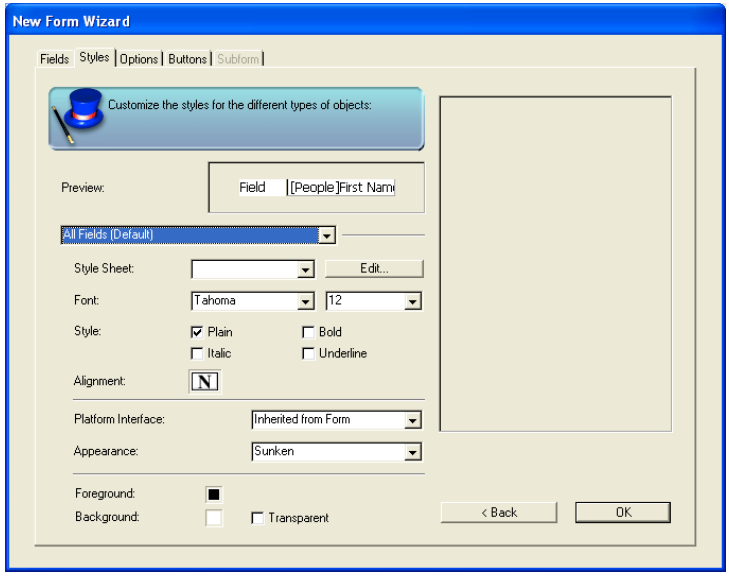
This section describes the advanced options available for creating Detail forms. For more information about the creation of List forms, refer to Chapter 7, "Output Displays and Reports" on page 477.

The Advanced screen contains the following pages:

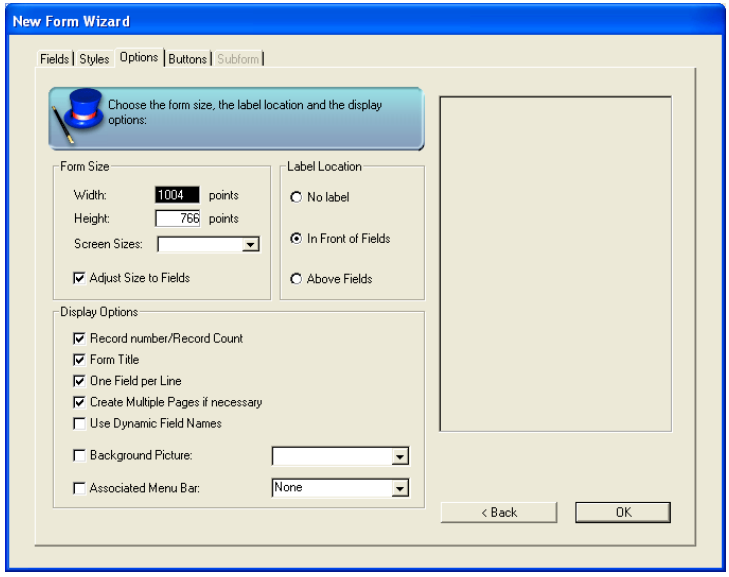
- **Fields page** Choose form fields. This page is similar to the Basic screen of the Form Wizard.



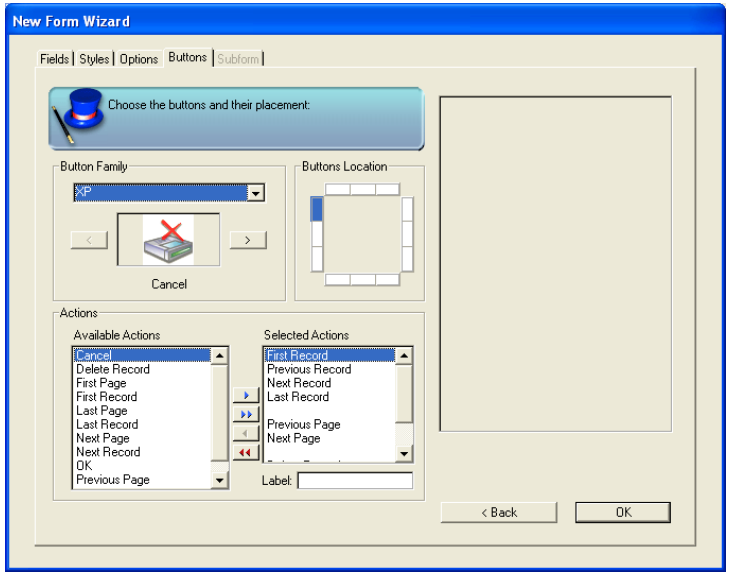
- **Styles page** Set the appearance of form objects and their labels. You can specify font attributes, foreground and background colors, platform interface, and appearance of decorative rectangles surrounding fields and field labels. You can also define or apply Style Sheets to specify font attributes.



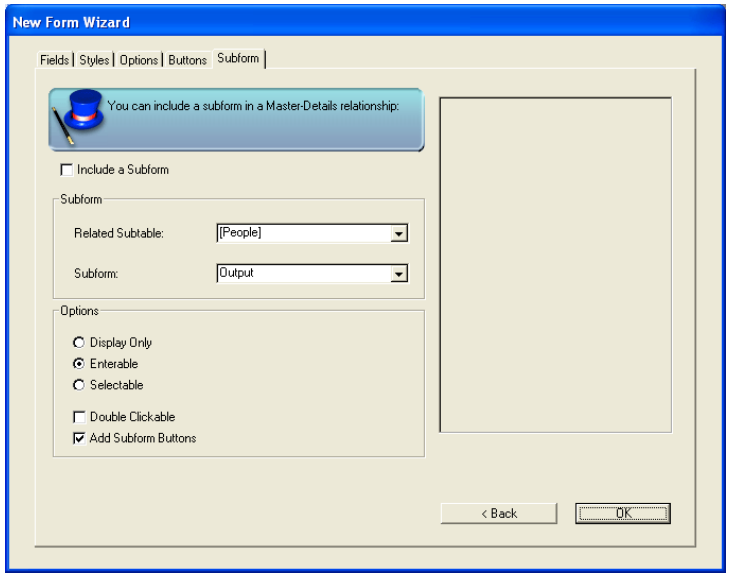
- **Options page** Set the form size, placement of field labels relative to fields, associate a background image, associate a menu, add a title and/or record count or use dynamic field names.



- **Buttons page** Choose a set of buttons to be added to the form, assign automatic button actions, set the placement of the buttons on the form, and (optionally) label the buttons.



- **Subform page** Add a subform from a related Many table or subtable to the form.



Note The Subform page is enabled only if there is a related many table related to the master table.

Adding Fields

Use the Fields page to add fields to the form.

You add fields to the form in exactly the same way as on the Basic screen of the Forms Wizard. This functionality is duplicated for users who want to skip the Basic screen and go directly to the Advanced options screen. For a complete description of the process of adding fields to the form, see the section "Selecting Fields for the Form" on page 237.

If you have already added fields to the form using the Basic screen, you can modify your selections using the Fields page. For more information, see the sections "Reordering Fields" on page 240 and "Removing Fields" on page 242.

Customizing the Appearance of Form Objects

The Styles page of the Forms Wizard lets you customize the appearance of the following types of objects:

- Related fields
- Non-enterable fields
- Mandatory fields
- Indexed unique fields
- Group box text
- The form title
- Information: Form elements that provide information using internal variables, such as page number, record number, and records in selection. Several of the default templates add such variables automatically.
- Check boxes and radio buttons
- Default: Fields and field labels not specified by other items in the Type drop-down list.

  For fields, you can specify the appearance of the field and the field's label separately.

  For each object type, you can customize the following properties:

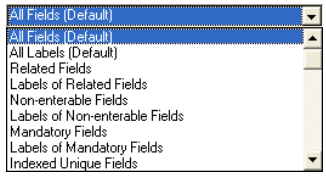
- Font attributes
- Platform interface
- Appearance of decorative rectangles surrounding the object
- Text justification
- Foreground and background colors.

▶ To customize the appearance of the form's objects:

1 Click the Styles tab in the Advanced options screen.

The Styles page appears.

2 **Choose the type of object whose appearance you want to customize from the drop-down list at the top of the window.**



When you make your selection, the preview area changes to show a preview of the type of object you selected. For each type of field, the drop-down list allows you to choose either the field area itself or its label; the preview area shows preview images for both the field and the field label.

3 **Choose the desired font, font size, and font style.**
OR
Choose a style sheet or click Edit to create a style sheet.

For information on creating and using styles, see "Using the Style Sheet Editor" on page 259.

The preview area reflects any modifications made.

4 **Choose the desired text alignment.**

The following choices are available in the form of icons:

- **Default (N)** Right-aligned numbers and left-aligned text, dates, and times,
- **Left**, **Right**, or **Centered** alignment.

For more information on alignment options, see the section "Using the Style Sheet Editor" on page 259.

5 **Choose the Platform Interface for the object from the Platform Interface drop-down menu.**

Your choices are:

- **Inherited from Form** Platform interface for the object where the label is the same as the one selected for the form. The form platform interface is defined in the form properties window. For information on setting the Platform Interface for the form, see "Setting the Platform" on page 292.
- **Automatic** The platform used is the one defined in the "Auto Platform" pop-up menu for the current platform in the Preferences. For more information on the Automatic option, see the paragraph "Look Page," page 110.
- **MacOS 7** The object will be displayed as a MacOS 7 object.
- **Windows 3.11, NT 3.51** The object will be displayed as a Windows 3.11 or NT 3.51 object.
- **Windows 95/98/2000, NT 4** The object will be displayed as a Windows 95/98/2000 or NT 4 object.
- **Mac OS 9** The object will be displayed as a MacOS 9 object.
- **Mac Theme** Under MacOS 9 and X, the object is displayed in relation to the defined theme in the "Appearance" panel (if this function is available). Under Windows, the object is displayed as a Windows 95/98/2000 or NT 4 object.

For complete information on how the Platform Interface options affect the appearance of objects, see the section "Platform Interface," page 98.

6 Choose the Appearance of decorative rectangles surrounding the object or label.

Your choices are:

- None
- Plain
- Dotted
- Raised
- Sunken
- Double.

For illustrations of the effects of these choices on various object types, see the sections "Fields and Field Labels" on page 258 and "Buttons" on page 421.

7 **Choose a foreground and background color or click the Transparent check box for automatic background (Optional).**

The foreground color is the color of text in an area. The background color is the color of the area itself. For more information, see the section "Foreground and Background Colors" on page 344.

8 **Choose another object type from the drop-down list and repeat steps 3 to 7.**

9 **When you are finished, click another tab to customize another aspect of the new form.**
OR
If you are finished with all Advanced pages, click OK to generate the new form.

Customizing Buttons on the Form

Detail forms use buttons that allow the user to save and cancel changes to a record, move from one record to another (first record, last record, next record, previous record), or one page to another in a multi-page form, add or delete records in a subform, or delete the current record. With the Buttons page in the Advanced options screen, you can choose a button design, choose the desired button actions, specify the position of the buttons on the form, and label each button.

Note In the Form editor, you can add, delete, or reposition individual buttons and attach a method to a button that specifies its action when clicked. For more information, see "Buttons" on page 421.

Automatic Button Actions

4th Dimension provides a set of built-in button actions. When you assign a built-in button action to a button, you don't need to write a method to specify what happens when a user clicks the button.

The built-in button actions are:

- **OK** Save a new record or save changes to an existing record,
- **Cancel** Discard the new record or discard changes to an existing record,
- **Next Record**, **Previous Record**, **First Record**, **Last Record** Save the current record and display the desired record,

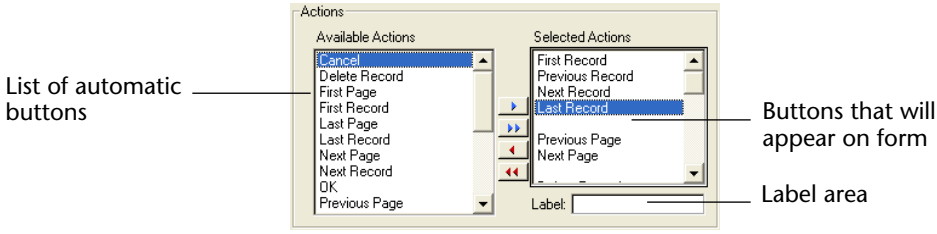
- **Next Page**, **Previous Page**, **First Page**, **Last Page** Display the desired page in a multi-page form,
- **Delete Record** Delete the current record from the database (a confirmation dialog box appears).

Note When you insert a subform, 4th Dimension can automatically insert three additional subform buttons if you click the **Add Subform Button** check box in the Options area of the Subform page. The subform buttons are: **Open** (enables the user to add a new record by opening a Detail form), **Add** (enables the user to add a new record by entering data directly into the subform), and **Delete** (delete the currently selected record in the subform).

You can assign other predefined actions to buttons. These actions are available when you create a form using the Form Wizard or when you modify a form using the Form editor.

For more information, refer to the paragraph "Automatic Button Actions," page 250.

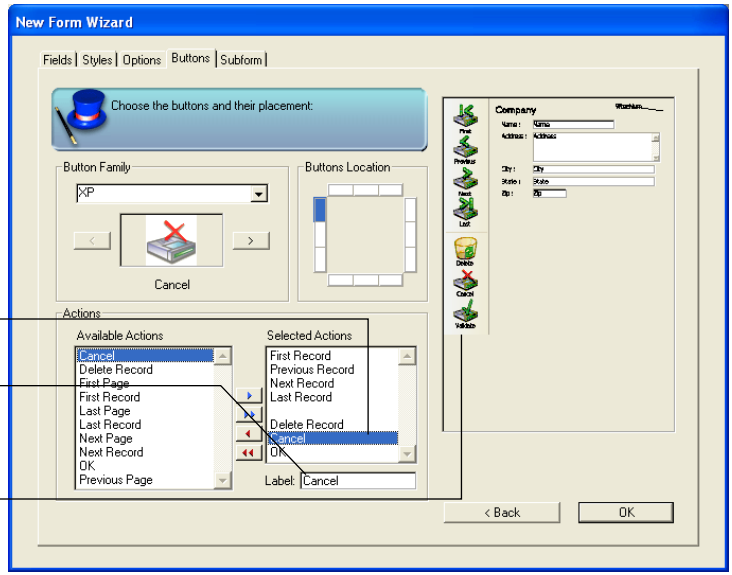
The default buttons are listed in the Actions area of the page.



You can select and deselect automatic buttons in the same way that you can add or delete fields from the form on the Fields page. The buttons that you add to the Selected Actions area appear on the form.

If you want to modify the default label of a button (the template chosen must include labels), highlight the button in the Selected Actions list and enter a new label in the Label area.

The following illustration shows a Cancel button whose label has been modified.

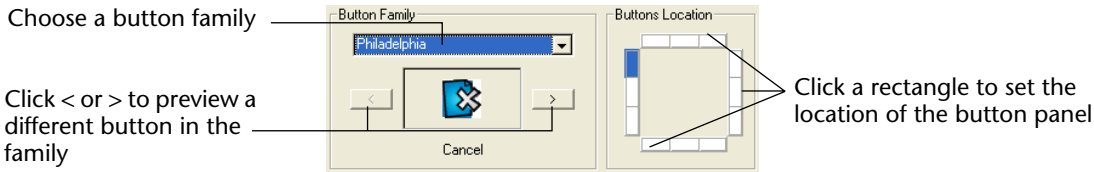


After entering the label, press **Tab** or click another button in the Selected Actions list. The label you entered is then displayed in the preview area, as shown above.

Note • You must select a template with labels on the main page of the wizard in order for the button family to display the default labels in the preview area. However, it is still possible to enter labels even when the button family does not display them by default.

• Button tips are independent of button labels. If you want to assign a tip to a button, you can do so using the Object Properties palette in the Form editor. For more information, refer to "Adding Help to a Field or Object" on page 393.

The Button Family and Button Location areas let you choose the style and location of the buttons.



Choose a family from the Button family drop-down list and click the > or < buttons to preview each button.

► To customize the buttons added to the form:

1 **Click the Buttons tab to display the buttons page in the Advanced screen.**

2 **Choose a button style from the "Button family" drop-down list.**
The preview area shows your selected button family.

3 **In the "Available Actions" area, click each button action that you want to include on the form or click the Append button .**
If you want to label a button, highlight it in the Selected Actions area and enter a label in the Label area.

4 **Select the location of the button palette by clicking on a rectangular area in the "Button Location" area.**
The preview area indicates the effects of your choice.

5 **When you are finish adding buttons and button actions to the form, click another tab to customize another aspect of the form.**
OR
Click OK to generate the form.

Setting the Form Size

The Options page of the Advanced screen lets you specify the form size. This section allows you to adjust the form size or set the form to a fixed size either by entering in its maximum width and height or by selecting a screen size. You can also combine the two settings.

The Screen Sizes drop-down list gives you the following choices:

- Automatic (the form size adapts to the current screen)
- Low Resolution 640x480
- Medium 800x600,
- Standard 1024x768,
- High resolution 1280x1024,
- Large screen 1600x1200.

Note The values correspond to the *width* x *height* ratio, expressed in pixels.

When you enter a screen size or choose a size from the drop-down list, the preview area changes to reflect your selection. The Form Wizard will try to adjust field and object placement on the form so that all the form objects will fit in the selected screen size. If the **Create Multiple Pages if necessary** option is selected and 4th Dimension cannot make all the fields fit in one page, it will generate multiple display pages to fit all the fields on the form. For more information about that option, refer to "Form Display Options" on page 254. If the Form Wizard generates multiple pages, it places buttons, the form title, and decorative rectangles on the background page (page 0).

Adjust Size to Fields If you check this box, the Form Wizard will shrink the background items around the fields so that less blank space is left.

Field Label Placement

The Field Labels area in the Options screen allows you to control where a field label is placed in relation to the field. If you want labels, they can be placed either in front of or on top of the fields.

Form Display Options

The Display Options area on the Options screen lets you add several optional elements to the form and set additional options. Your choices are:

- **Record Number/Record Count** Adds a 4th Dimension variable to the form that display the current record number and the total number of records. That variable is named vRecNum. It can be edited in the Form editor as any other variable.
- **Form Title** Adds the name of the table as the title of the form above the fields.
- **One Field per Line** Check to arrange the fields vertically. If this option is not checked, the Form Wizard will try to arrange fields in rows.
- **Create Multiple Pages if necessary** Check to have the Form Wizard create extra pages automatically if the fields do not fit on one page. If you use this option, the Form Wizard places the appropriate objects on the background page.
- **Use Dynamic Field Names** When this option is selected, field and table names are inserted in the form as dynamic references. It ensures that the field and table labels will reflect any changes to the field or table name. Table or field names can be modified in the Structure editor or using the Table name or Field name commands. For more information, see "Inserting Dynamic Table and Field Names" on page 370.

- **Background Picture** Click the Background Picture check box and choose a background from the drop-down list to add a background to the entire form.

Note This option does not allow you to define a background picture for a form published on the Web. For information about how to set a background picture for a Web form, refer to "Defining a Background Picture for Web forms" on page 348.

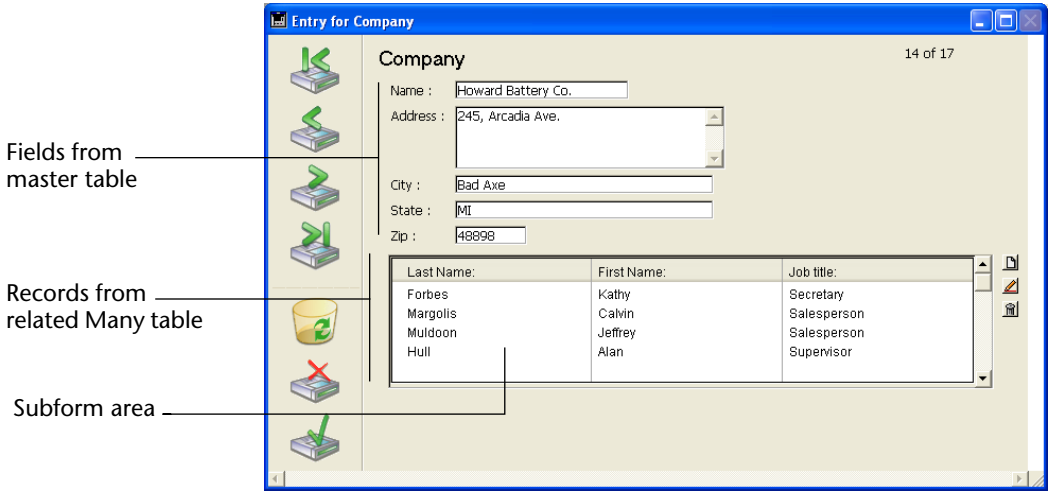
- **Associated menu bar** Enter the number of a menu bar that you want to associate with the form[1]. For me information, please refer to "Assigning a Menu Bar to a Form" on page 295

Adding a Subform to the Form

When you want to use fields from a related Many table or from a subtable, add a subform to the form. The subform lists several records at once.

Using a subform allows you to view the related records or subrecords. You can also enter information into records and subrecords that are displayed in the subform.

The figure below shows a Detail form with a subform during data entry.

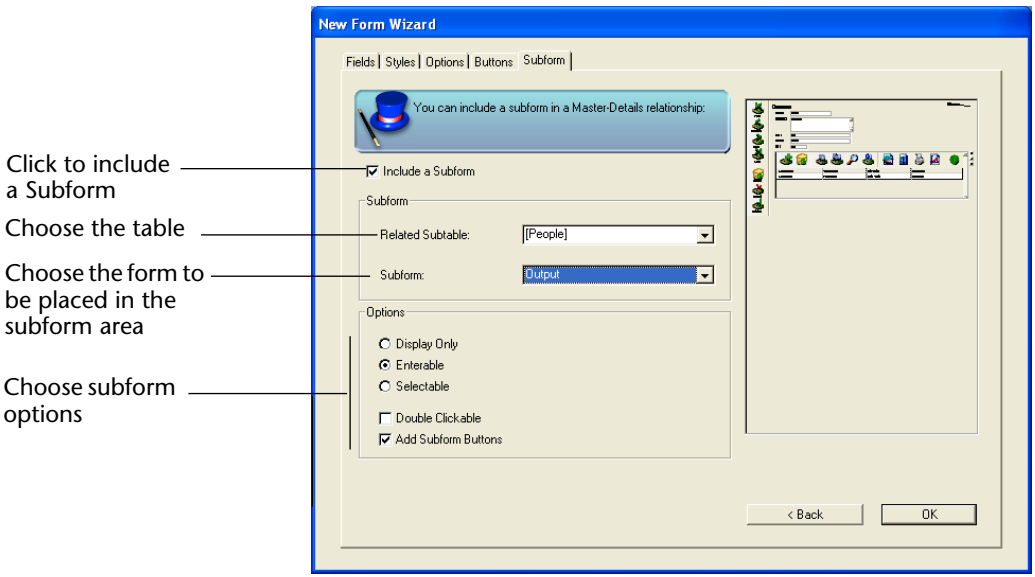


1. You can also associate a menu bar with a form using the Form editor.

You can display fields from a subtable, a related Many table, or an unrelated table in a subform. If you include fields from a related Many table, the relation determines which records are displayed. If you include fields from an unrelated table or from a table with a manual relation, by default the current selection of records from that table is displayed. You can also control the selection of records using a method.

You can include subfields from a subtable in a subform. The subrecords for the parent record are displayed in the subform. Only those subrecords that belong to the parent record are displayed.

The Subform page[1] of the Advanced screen lets you add a subform to the form, specify subform options, and add buttons to allow users to work with the subform.



► To add a subform to the form:

1 Click the Include a Subform check box.

If there is only one candidate related Many table or subtable, 4th Dimension adds the subform to the form. It appears in the preview area and the related Many table or subtable is shown in the Subform area.

1. The Subform page is available only if the master table contains a subtable or has a related Many table.

2 **If necessary, choose the desired related Many table or subtable from the drop-down list.**

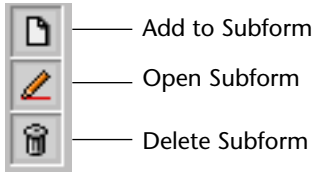
3 **Choose the desired form to use as the subform from the drop-down list.**

4 **Choose any desired options from the Options area, then click OK.**

For information on subform options, see the section "Data Entry Options for Subforms" on page 470.

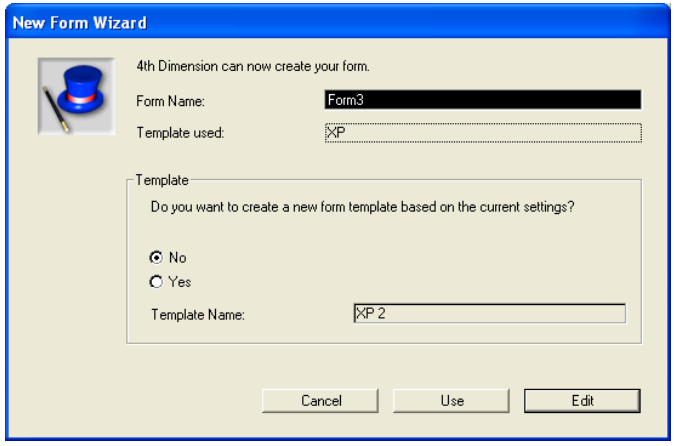
The **Add Subform Buttons** check box adds a standard subform button panel to the form. The button panel contains **Add to Subform**, **Open Subform**, and **Delete Subform** record buttons.

Here is the standard subform button panel:



Creating the New Form

When you have finished specifying all properties of the new form, click **OK** on any page to create the new form. When you click **OK**, the following dialog box appears:



To create the new form, click either **Use** to switch to the User environment to test the form or **Edit** to open the new form in the Form editor.

Creating a Form Template

The Accept Form dialog box gives you the option of creating a new form template using the current Advanced settings. If you create a form template, its name will be added to the Form Template drop-down list in the Basic screen of the Form Wizard. The form template is saved in addition to the form itself.

► To create a form template:

1 **Click the Yes radio button in the Template area and enter a name in the Template Name area.**

2 **Click either the Use or Edit buttons.**

Edit opens the new form in the Form editor, ready for further customization. **Use** switches to the User environment so that you can begin using the new form.

If you click **Use**, you can return to the Design environment and open the form in the Form editor at any time.

Setting the Appearance of Form Objects

The Styles page of the Form Wizard lets you specify the platform interface, appearance, and font attributes of fields and field labels, text, and check boxes and radio buttons. This section describes the available options.

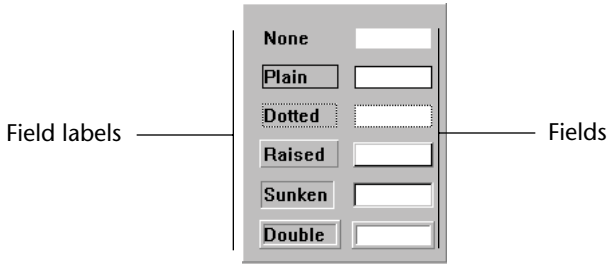
Fields and Field Labels

The **Appearance** drop-down list in the Styles page lets you customize the look of the fields you place on the form. Your choices are:

- **None** All objects appear with their normal graphical settings according to the selected Platform Interface and user choices.
- **Plain** The non-framable objects appear with their normal graphical settings according to the selected Platform Interface. The framable objects appear with a solid 1-pt frame.
- **Dotted** The non-framable objects appear with their normal graphical settings according to the selected Platform Interface. However, the dotted appearance overrides the selected pattern. The framable objects appear with a dotted 1-pt frame.
- **Raised** All objects (or their frames) appear with a 3D raised effect.
- **Sunken** All objects (or their frames) appear with a 3D sunken effect.

- **Double** Under MacOS, the objects or their frames appear with a double-line: two 1-pt solid lines separated by one pixel. Under Windows, the objects or their frames appear with one black and one white lines offset by one pixel.

The following illustration compares the six appearance settings.



You can customize the appearance of fields and their labels separately.

After you create the form, you can set the Appearance of individual objects using the Object Properties window in the Form editor. For more information, see the section "Changing the Appearance of Objects" on page 334.

Using the Style Sheet Editor

The Style Sheet editor lets you save sets of font attributes — font, font size, and style — as a named style sheet. The style sheet can then be used to specify font attributes in the Styles page of the Forms Wizard or the Font page of the Object Properties window.

Style sheets modify the appearance of the objects that use them. For example, changing the font size of a style sheet will change the font size for all the objects using this style sheet.

Each named style sheet saves separate sets of font attributes for each platform interface supported by 4th Dimension. For example, the MacOS platform interface could use Geneva as the font, while the Windows platform interfaces could use MS Sans Serif. Similarly, the font sizes can be specified separately for each platform interface.

Creating a Style Sheet

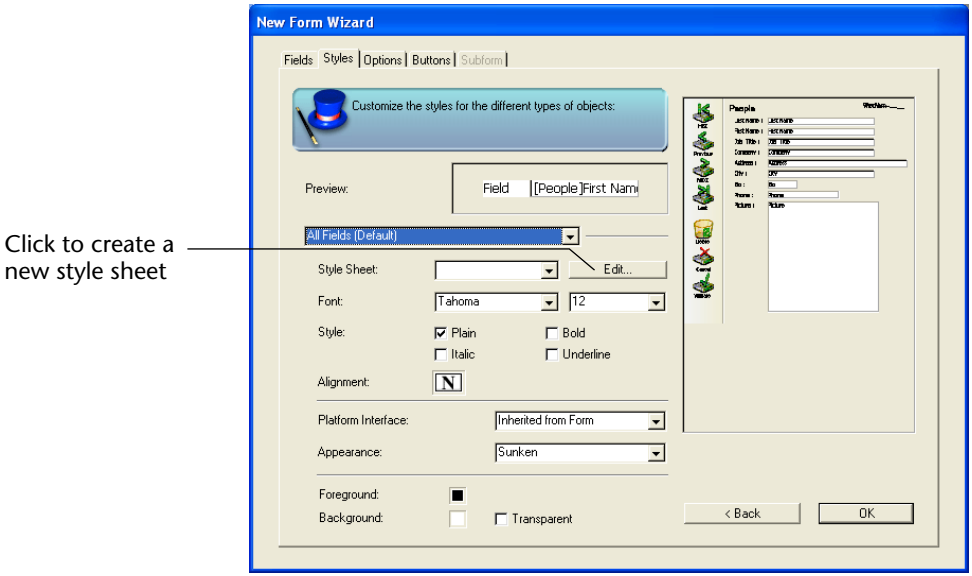
You can create style sheets from several places in 4th Dimension:

- From the Style sheet page of the Preferences dialog box (Interface theme). For information, see the section “Style sheet Page,” page 114.
- From the Styles page in the Advanced screen in the Form Wizard (**Edit...** button).
- From the Font page of the Object Properties window (**Edit** button).
- In the Property list (**[...]** button located next to the style sheet drop-down list).

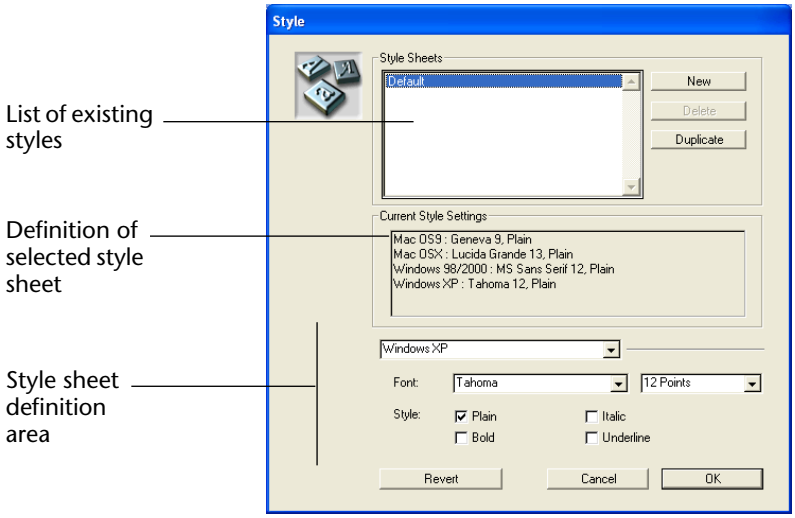
The dialog box used for creating or modifying style sheets is the same. No matter what location it is called from, the list of style sheets is the same.

► To create a style sheet:

1 Click the Style Sheet Edit... button in one of the dialog boxes mentioned above.



The Style Sheet Definition dialog box appears:

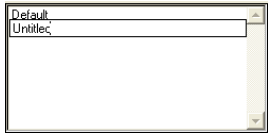


2 **Click New.**

A new style sheet appears in the list of existing styles. Its default name is "Default."

3 **To rename the new style, hold down the Command key (MacOS) or the Ctrl key (Windows) and click on its name.**
OR
Double-click the style's name.

The text becomes editable.



4 **Rename the style sheet and press Tab or click anywhere outside the entry area to save your changes.**

5 **In the Style Sheet definition area, choose a platform whose font attributes you want to define.**

4th Dimension identifies four platform families:

- **Mac OS9**: indicates any Mac OS system prior to Mac OS X.
- **Mac OSX**: new version of Mac OS.
- **Windows 98/2000**: indicates any Windows system prior to Windows XP.
- **Windows XP**: new version of Windows.

6 **Choose the desired font, font size, and font style options.**

The Definition area changes to reflect your changes.

7 **Repeat steps 5 and 6 for each platform (optional).**

When you are finished defining the style sheet, you can click **New** to create another style sheet or click **OK** to save the new style sheets and put away the Style Sheet Definition dialog box.

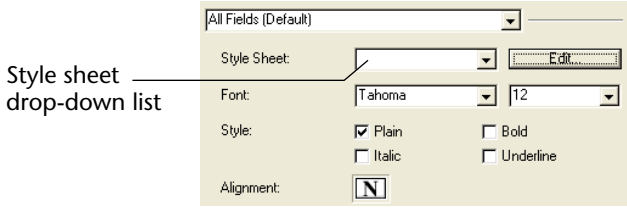
Note If you like, you can begin a new style sheet definition by duplicating an existing style sheet by clicking **Duplicate** rather than by clicking **New**. This will allow you to avoid having to respecify the properties of the new and old styles will have in common.

The **Revert** button allows you to cancel **all** the changes you made since you opened the **Style** dialog box (creation, modification or deletion of style sheets). Clicking it reverts the Style dialog box to the state it was in when it was opened.

When the Styles page of the Form Wizard reappears, the new style sheet names appear in the Style Sheet drop-down list. You can then specify font attributes by choosing a style sheet rather than by making font, font size, and style selections.

Applying a Style Sheet

To apply a style sheet to an object, choose the name of the style sheet from the Style Sheet drop-down list.



Your selection sets the font, font size, and font style attributes for the currently selected label or object.

Setting the Current Input and Output Forms

Each table has one current input form and one current output form. The input form is used for entering and modifying records, and the output form is used to list records. Usually, you use a Detail form for input and a List form for output.

You can change which form to use for input and output at any time. You can change these designations in both the Design and User environments as well as using the commands INPUT FORM and OUTPUT FORM.

► To designate input and output forms:

1 **Choose Edit Form from the Design menu.**
OR
If the Explorer is already displayed, click the Forms tab to display the Forms page.
4th Dimension displays the Forms page of the Explorer.

2 **If the Explorer is not already displaying the preview area, click the preview area icon.**

3 **Select the table whose form you want to set and expand it to view the forms for that table.**

4 **Select the form you want to use for input.**
The preview area shows the form.

5 **Click the Input Form check box below the preview area.**
The **I** character (for **I**nput) is then displayed next to the form's name. This indicates that the form is the current input form.

6 **Select the form you want to use for output.**
Its preview appears in the preview area.

7 **Click the Output Form check box.**
The **O** character (for Output) is then displayed next to the form's name. This indicates that the form is the current output form.

Note You can also use the contextual menu of the Explorer (**right-click** under Windows or **Control+click** under MacOS on the form name). The **Input Form** and **Output Form** commands let you specify the corresponding forms.

You can also designate the same form as the Input and Output form. In this case the character **B** (for **B**oth) will be displayed next to it.

Deleting a Form

You can delete any form that is not currently designated as an input form or an output form (or both). The **Delete** button is disabled when you select the current input or output form.

► To delete a form:

1 **Choose Edit Form from the Design menu.**
OR
If the Explorer is already displayed, click the Forms tab to display the Forms page.
4th Dimension displays the Forms page of the Explorer.

2 **Select the table that contains the form you want to delete and expand it to view the forms for that table.**

3 **Select the form you want to delete.**
When you select a form that is not the current input or output form, the **Delete** button becomes active.

4 **Click the Delete button.**

Note You can also use the **Delete Form** command in the contextual menu of the Explorer (**right-click** under Windows or **Control+click** under MacOS on the form name).

4th Dimension asks you to confirm the deletion.

5 **Click the OK button.**
4th Dimension deletes the form.

Renaming a Form

You can rename a form using either the Explorer or the Form editor.

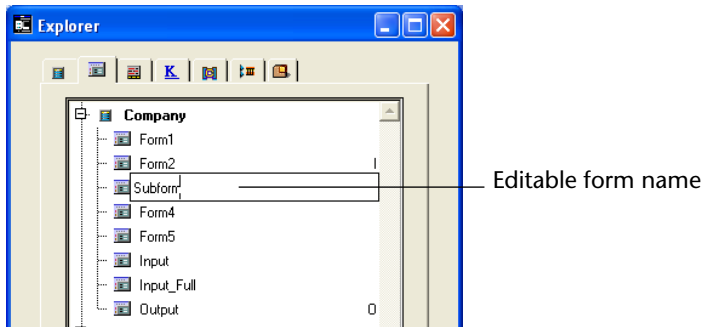
► To rename a form using the Explorer:

1 **Choose Edit Form from the Design menu.**
OR
If the Explorer is already displayed, click the Forms tab to display the Forms page.
4th Dimension displays the Forms page of the Explorer.

2 **Select the table that contains the form you want to rename and expand it to view the forms for that table.**

3 **Hold down the Command key (MacOS) or the Ctrl key (Windows) and click the form name you want to rename.**

The name becomes editable.



4 **Type a new name.**

5 **Press Tab or click anywhere outside the entry area to save your changes.**

4th Dimension changes the name of the form.

Note Changing a form name can invalidate any methods or formulas that use the previous form name. Each of these items has to be updated in order to function.

4D Server The form name is changed on the server when the user clicks outside the entry area to save the new name. If more than one user is modifying the form name at the same time, the final form name will be the name specified by the last user to save the name.
You may want to specify a form owner so that only certain users can change the form's name. For more information about specifying owner privileges for a form, refer to the section "Setting Form Access" on page 291.

► To rename a form using the Form editor, refer to the paragraph "Setting Form Properties," page 289.

5 Form Editor Basics

When you create a new form with the Form Wizard, you can choose many customization options. Using templates, you can control the font, font size, and style of text, control the appearance of fields and field labels, and add a set of automatic buttons.

This is only the beginning though, since 4th Dimension provides a full-featured Form editor that allows you to modify your form until you achieve the effect that you want. With the Form editor, you can create and delete objects, manipulate objects directly, and set form and object properties.

This chapter provides an introduction to the Form editor. It covers:

- The Form editor window
- The Form editor Tools palette
- The Form editor menus
- Tools for displaying and editing form and object properties

The following operations are explained in detail:

- Opening a form in the Form editor,
- Setting form properties,
- Creating and managing objects,
- Moving and resizing objects,
- Grouping and ungrouping objects,
- Aligning objects,
- Copying objects,
- Layering objects,

- Creating text areas on a form,
- Adding a picture from the Picture Library to the form,
- Creating a multi-page form,
- Changing the data entry order of the fields.

Chapter 6 covers fields and other active objects in detail.

4D Server Object locking occurs when two or more users attempt to modify the same form simultaneously. If a user is modifying a form in the Design environment, the form is locked. Other users cannot modify that same form until the first user frees the form by closing it.

Using The Form Editor

This section provides an overview of the Form editor and describes the tools available for editing forms. Subsequent sections describe in detail how to work with the Form editor's tools.

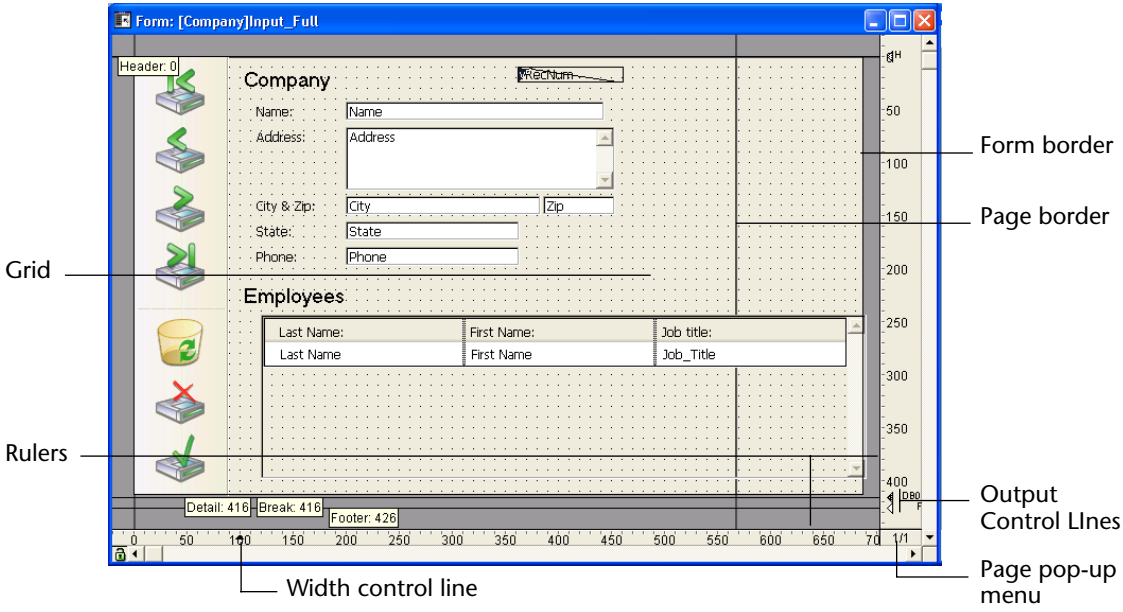
This section introduces the four components of the Form editor:

- The Form editor window,
- The Tools palette,
- The Form editor menus,
- Palettes for editing and displaying form and object properties.

Form Editor Window

4th Dimension's Form editor is a powerful object-oriented graphics editor that includes special features for working with form objects.

The following illustration shows the Form editor window:



The Form editor displays each form in its own window. You can have several forms open at the same time. The rulers on the side and bottom help you position objects in the form. You can change the units the ruler uses so that it measures in inches, centimeters, or pixels.

Note The output control lines are relevant only for output forms. They control the information that is listed and define header and footer areas. The label width triangle on the horizontal ruler controls the width of a label when you create a form for printing mailing labels using the PRINT LABEL command. For information on working with output control lines, see the paragraph "Moving Output Control Lines" on page 483.

The Form editor menus added to the 4th Dimension menu bar provide menu commands that allow you to change the data entry order of the fields, control interface elements and form pages, manage objects, and change the font and style for displaying information.

In addition to those menus, the Form editor also includes contextual menus that allow you to quickly access numerous actions. For more information, see the section "Form Editor Menus" on page 277.

Showing/Hiding Elements in the Form Editor

You can show or hide most interface elements in the Form editor. This feature allows you to show only the elements that you need to create or view in a form or only the tools that you want to use. This option is always applied to the Form editor's current window.

For example, it is useful to show the output control lines when you are working on an output form.

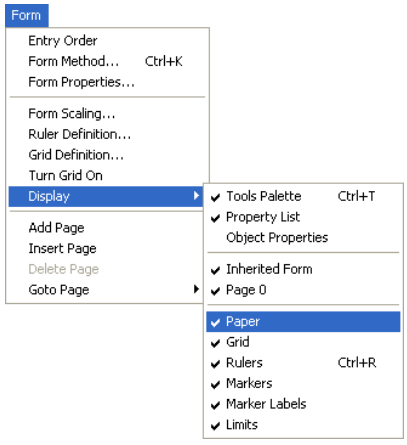
► To show or hide an element in the Form Editor:

1 Choose Display from the Form menu.
OR
Use the Display command in the contextual menu that appears in the Form Editor's window:

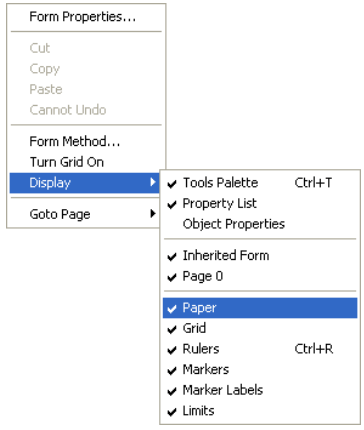
- Under Windows, click with the **right mouse button** (without clicking on an object).
- Under MacOS, **Control+click** (without clicking on an object).

A hierarchical submenu appears listing all the elements that you can show or hide:

Form menu



Contextual menu



A check mark placed next to the element indicates that it will be shown. To hide an element, select the element so that the check mark disappears.

2 Select an element that you want to show or hide.

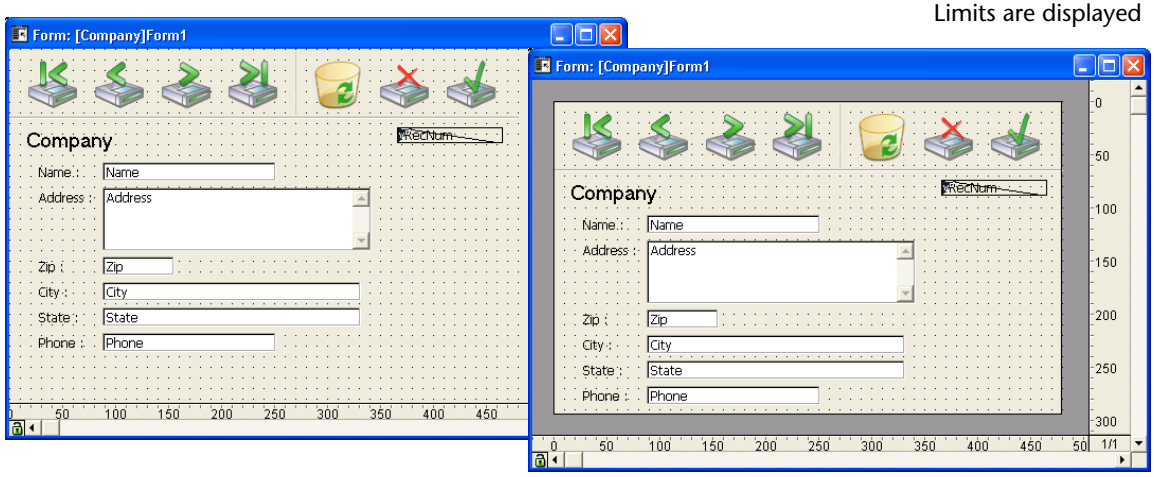
Here is a description of the commands in this menu:

- **Tools Palette**: shows or hides the Form editor's Tools palette (see the section "The Tools Palette" on page 273).
- **Property List**: shows or hides the Property List. This list is described in the section "Comparing the Property List and property windows" on page 286. The Property List is an alternative to the Object Properties window. You can toggle between the Property List and the Object Properties window; you cannot display both simultaneously. For more information on both the Object Properties window and the Property List, refer to "Displaying and Setting Form and Object Properties" on page 282.
- **Object Properties**: shows or hides the Object Properties window. This window can be replaced by the Property List (see above).
- **Inherited form:** shows or hides inherited form objects (if there is an inherited form) in the current page of the form. For more information, please refer to "Inherited Forms" on page 355.
- **Page 0**: shows or hides the objects from page 0 in the form's current page. This option allows you to distinguish between the objects on the form's current page and those on page 0. For more information about pages in forms, refer to the section "Creating a Multi-page Form" on page 350.
- **Paper**: shows or hides the borders of the printing page, which are shown as gray lines. This option can have no apparent effect when the **Limits** (see below) option is selected. If the size of the form is smaller than the printing page, the page's borders are shown outside of the form's viewing area and therefore do not appear.
- **Grid**: shows or hides the grid that is used as a guide when aligning objects. In previous versions of 4th Dimension, this grid was "invisible" and its appearance was related to the ruler unit that was chosen. The grid that appears in the form's background is independent from of the ruler; however, you can still use it to align objects on the form. You can define the grid's unit in the Grid Definition dialog box, which you can access by choosing **Grid Definition** in the **Form** menu. For more information on the grid, refer to "Using the Alignment Grid" on page 320.
- **Rulers**: shows or hides the rulers in the Form Editor's window as well as the pop-up menu that allows you to navigate the form's pages.

- **Markers**: shows or hides the output control lines and associated markers that show the limits of the form's different areas.
- **Marker Labels**: shows or hides the marker labels, available only when the output control lines are displayed. For more information, please refer to the paragraph "Moving Output Control Lines", page 483.
- **Limits**: shows or hides the form's limits. When this option is selected, the form is displayed in the Form editor as it appears in User mode.

Limits are not displayed

Limits are displayed



Note The **Size Based on**, **Hor margin** and **Vert margin** settings of the form properties affect the form's limits. When using these settings, the limits are based on the objects in the form. When you modify the size of an object that is located next to the form's border, it is modified to reflect that change. For more information on form properties, refer to "Setting Form Properties" on page 289.

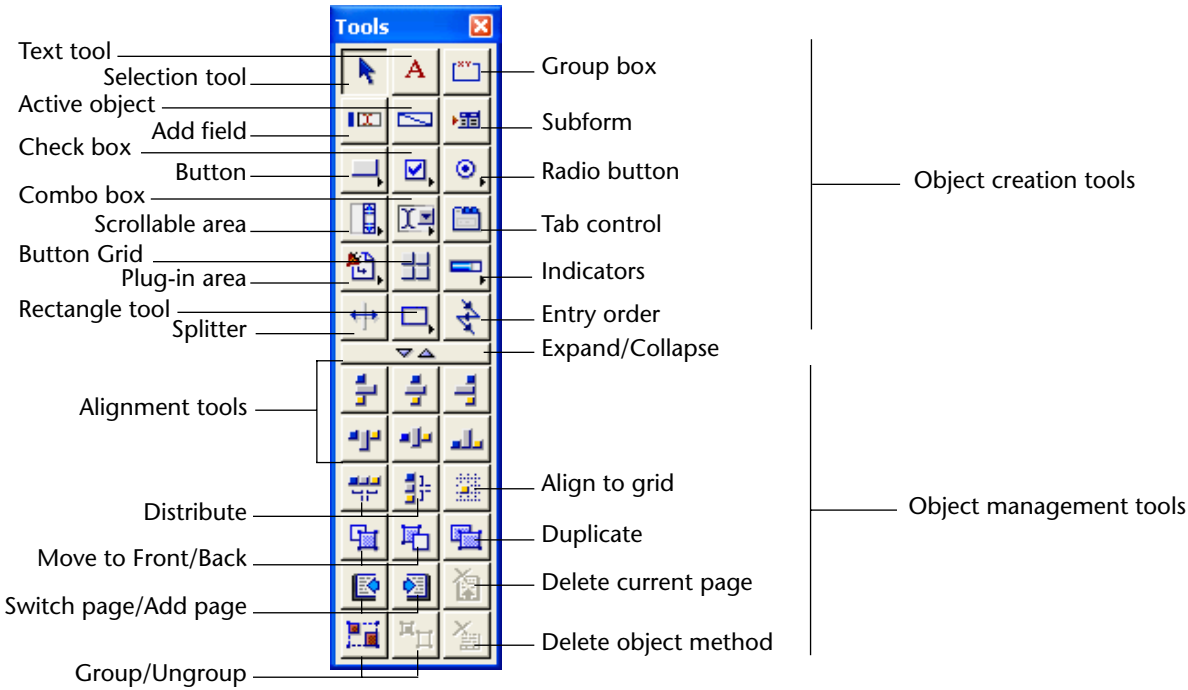
The Tools Palette

The Tools palette provides a collection of tools for creating and manipulating form objects.

► To display the Tools palette of the Form editor:

1 **Choose Display then Tools Palette in the Form menu or in the contextual menu of the editor (click outside any object).**
OR
Press Ctrl+T (Windows) or Command+T (MacOS).

The Tools palette consists of two main areas:



The upper area of the Tools palette contains object creation tools. The lower area of the Tools palette contains object management tools. You can hide the lower area of the palette by clicking the button located in the middle of the Tools palette.

When you hide the object management area in the Tools palette, it appears as follows:

To display the lower part of the Tools palette, click the  button.

Object Creation tools

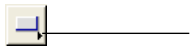
You can create an object by:

- Dragging an object from the Tools palette to the form
- Selecting a tool and drawing the object's shape on the form.

Once the object is created, it is selected and assigned default values. Depending on the display settings you selected, either the Property list or the Object Properties window is also displayed. For more information on object types and properties, refer to "Types of Active Objects" on page 419.

After creating an object, you can modify its type and/or properties in the Object Properties window or in the Property List.

Some buttons in the Tools palette are drop-down lists that contain specific object types. These buttons are indicated by a small triangle:



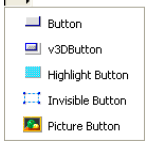
Make your selection from the drop-down list associated with such a button.

▶ To select a variation of an object type:

- **Click the object button and do not release the mouse button (long click) until the drop-down list appears.**
  OR
  Under Windows, right-click the object button.
  Under MacOS, click the object button while pressing the Control key.

A menu is then displayed, which allows you to select an object. The menus are shown below:

Button

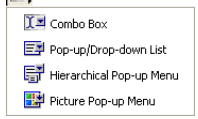


Check box

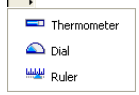
Radio button

Scrollable area

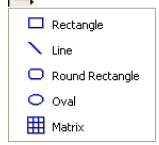
Combo box



Indicator



Shapes



Note Depending on the plug-ins that are installed in your database, the **Plug-in Area** button may also be associated with a drop-down list.

Here is a brief description of the tools in the Tools palette:

- **Selection tool** Used to select, move, and resize objects in the form. For more information, see the section "Managing Form Objects" on page 302.
- **Text tool** Used to create or edit text on the form for labels, titles, instructions, and so on. For more information, see the section "Creating and Editing Text Areas" on page 336.
- **Group Box tool** Used to create a box with a built-in label in the upper left corner.
- **Add Field tool** Used to add a field to a form. For more information, see the section "Adding Fields to a Form" on page 367.
- **Active Object tool** Used to create variables or (optionally) other types of active objects. For more information, refer to "Enterable and Non-enterable Variables" on page 420.
- **Subform tool** Can be used to create an area that displays multiple records from another table or a subtable. For more information, see the section "Adding a Subform to the Form" on page 468.
- **Button**, **Check box**, **Radio button**, **Scrollable area**, **Drop-down list**, **Tab control**, **Plug-in area**, **Button grid**, **Indicator**, **Splitter tools** Used to create active objects. For more information, refer to "Active Objects on a Form" on page 412.
- **Graphic object tools** Used to create graphical objects such as circles, squares, rectangles, rounded rectangles, and matrixes. For more information, refer to "Creating Objects" on page 312. For more information on how to create and use a matrix, refer to "Duplicating on a Matrix" on page 459.

- **Entry order tool** Used to edit the entry order of the current form. When you click this tool, 4th Dimension displays the current entry order for the form, allowing you to modify it. For more information on the entry order, refer to "Data Entry Order" on page 358.

Object Management Tools

The following is a brief description of the Object Management tools in the Tools palette:

- **Alignment and distribution tools** These tools are used to align and distribute objects in the form. These tools may be disabled, depending on the objects that are selected (i.e., unless at least two objects are selected, these tools are not meaningful). For more information, refer to "Aligning Objects" on page 316.
- **Align to grid** Used to align objects on the grid that is defined for the form. For more information, refer to "Using the Alignment Grid" on page 320.
- **Layering tools** Used to move objects to the front or back when objects overlap. For more information, see the section "Layering Objects" on page 329.
- **Duplication tool** Used to duplicate the selected objects. For more information, refer to "Duplicating Objects" on page 326.
- **Page tools** Used to move to the background page, next or previous display pages, or to add additional display pages to the form. For more information, see the section "Creating a Multi-page Form" on page 350.
- **Grouping/Ungrouping tools** Used to group or ungroup a selection of objects. For more information, refer to "Grouping Objects" on page 315.
- **Object method deletion tool** Used to delete the object method assigned to the selected object. For more information, refer to chapter 8, "Creating Methods" on page 517.

Form Editor Menus

When a Form editor window is the active window, the following menus are added to the menubar:

- Form
- Object
- Font
- Style

In addition to the standard menus, the Form editor also includes contextual menus that you can select at any time.

The Form Menu

Use the **Form** menu to organize form elements. Some of its commands display a hierarchical submenu. The figure below shows the **Form** menu.

The following is a description of the menu commands in the **Form** menu, with a brief description of their use:

- **Entry Order** Used to create a custom entry order for data entry objects in an input form. When the entry order mode is selected, a check mark is displayed next to that menu command. For more information, see the section "Data Entry Order" on page 358.
- **Form Method** Selecting this menu command opens the form's method in the Method editor. If no method assigned to the form, the New Method dialog box is displayed. For more information, refer to chapter 8, "Creating Methods" on page 517.
- **Form Properties** Set or modify form properties. For more information, see the section "Setting Form Properties" on page 289.
- **Form scaling** Enlarge or reduce all form objects by a specified factor. For more information, see the section "Scaling a Form" on page 333.

- **Ruler definition** Used to set the scale of the form rulers. For more information, see the section "Using the Rulers" on page 311.
- **Grid definition** Used to define the scale of the form grid to which objects are aligned. For more information, see the section "Aligning Objects" on page 316.
- **Turn Grid On/Off** Used to turn on or off the grid to which objects can be aligned. For more information, see the section "Aligning Objects" on page 316.
- **Display** Used to display or hide interface elements such as palettes, rulers, control lines, and so on. The commands of this submenu are described in the section "Showing/Hiding Elements in the Form Editor" on page 270.
- **Add Page** Selecting this menu command adds a page after the last page.
- **Insert Page** Selecting this menu command adds a page before the current page.
- **Delete Page** Selecting this menu command deletes the current page.
- **Goto Page** Selecting this menu command displays a hierarchical submenu that lets you choose between the form's existing pages. Selecting a page number automatically displays the corresponding page. This command is equivalent to the navigation pop-up menu located in the lower right of the editor window when the rulers are displayed. For more information about managing multi-page forms, refer to the section "Creating a Multi-page Form" on page 350

The Object Menu

Use the **Object** menu to modify and manipulate form objects. Some **Object** menu commands display a hierarchical menu of choices. The following illustration shows the **Object** menu.

Here is a description of the menu commands in the **Object** menu, with a brief description of their use:

- **Line Width** Displays a hierarchical menu of line width choices for lines and borders. For more information, see the section "Line Widths" on page 341.
- **Fill** Displays a hierarchical menu of fill-pattern choices for objects. For more information, see the section "Fill Patterns" on page 342.
- **Border** Displays a hierarchical menu of border-pattern choices for objects. For more information, see the section "Border Patterns" on page 343.
- **Color** Displays a hierarchical menu of color choices for objects. For more information, see the section "Foreground and Background Colors" on page 344.
- **Move to Front** Used to move an object in front of all other objects. For more information, see the section "Layering Objects" on page 329.
- **Move to Back** Used to move an object in back of all other objects. For more information, see the section "Layering Objects" on page 329.

- **Up One Level** Used to move the object selection up one level to the front.
- **Down One Level** Used to move the object selection down one level to the background.
- **Group** Used to combine multiple objects in the form into groups that you can manipulate as a single object. For more information, see the section "Grouping Objects" on page 315.
- **Ungroup** Used to separate grouped objects into individual objects. For more information, see the section "Grouping Objects" on page 315.
- **Align to Grid** Used to align an object to an invisible grid in the form. For more information, see the section "Aligning Objects" on page 316.
- **Align** Selecting this menu command displays a submenu that allows you to choose between several alignment and distribution types. By selecting the Alignment submenu item, you can also display the alignment wizard. For more information about the use of the Alignment Assistant, refer to "Aligning Objects" on page 316 and "Distributing Objects" on page 322.
- **Duplicate** Used to duplicate objects. For more information, see the section "Duplicating Objects" on page 326.
- **Duplicate on Matrix** Selecting this menu command duplicates an object N-1 times, using a matrix object that contains N cells. This menu command is enabled only if an object and a matrix are selected and the object is contained in the upper left cell of the matrix. For more information, refer to "Duplicating on a Matrix" on page 459.
- **Duplicate many** Selecting this menu command displays the Duplicate many dialog box. This dialog box allows you to duplicate the same object several times and automatically assign each instance a numbered name. For more information, refer to "Duplicating Objects" on page 326.
- **Show Format**, **Show Resource**, **Show Name** Controls what is displayed on the form for objects whose text is actually stored in resources. Show Resource shows the STR# resource in which the text of the object is actually stored. These menu commands also allow you to display the references for field and table labels if they have been inserted using dynamic references. For more information, refer to the section "Working with Text Areas" on page 336 and section "Inserting Dynamic Table and Field Names" on page 370.

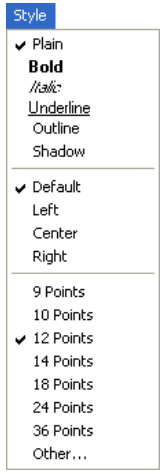
- **Object Method** Selecting this menu command opens the object's method in the Method editor. If there is no object method assigned to the object and you haven't chosen a default Method editor in the Preferences, 4th Dimension displays the New Method dialog box. For more information on this point, please refer to Chapter 8, "Creating Methods" on page 517.
- **Clear Object Method** Used to remove an object method from a selected object. To remove a method, select the object and choose **Clear Object Method** from the **Object** menu.
- **Insert an OLE Object** This command allows you to create a plug-in area that is designed to become an OLE area. The use of OLE areas is described in a separate manual named "OLE Areas." For more information on this point, please refer to the *4D OLE Tools* manual.

The Font Menu

Use the **Font** menu to apply a font to objects on the form and to specify a default font for text objects that you subsequently add to the form. The appearance of the **Font** menu depends on the screen fonts that you have installed on your system.
For more information, see the section "Changing the Appearance of Objects" on page 334.

The Style Menu

Use the **Style** menu to apply a text style, alignment, and font size to objects in the form or to specify these attributes as defaults for objects that you subsequently add to the form.



For more information, see the section "Changing the Appearance of Objects" on page 334.

You can set text attributes using either the **Font** and **Style** menus or the Fonts page of the Object Properties window.

Contextual menus

You can use contextual menus in the Structure editor. To use a contextual menu: Under Windows, click on an object or area with the right mouse button. Under MacOS, **Control+click** on an object or area.

With the exception of standard menu commands such as **Copy** and **Paste**, commands in contextual menus vary according to the current editor as well as the object on which you clicked:

- If you clicked a form object, the contents of the contextual menu are object-related: **Object Properties**, **Object Method**, **Color**, **Alignment**, **Border Line Style**, **Automatic Size**, **Automatic Action** (depending on the object's type) and **Level**. Depending on the object's type, additional commands may be displayed. For instance, **Attached Field** is displayed for a field, **List** for a hierarchical list.
- If you click outside any object, the contextual menu's items apply to the form: **Form Properties**, **Form Method**, **Turn Grid On**, **Display** (for more information on this menu command, refer to "Showing/Hiding Elements in the Form Editor" on page 270) and **Goto Page**.

Displaying and Setting Form and Object Properties

Both forms and form objects have properties that control access to the form, the appearance of the form, and the behavior of the form when it is used. Form properties include, for example, the form's name, its menu bar, and its platform interface. Object Properties include, for example, an object's name, its dimensions, its background color, and its font.

This section describes how to display and modify form and object properties. For more information on form properties, refer to "Setting Form Properties" on page 289. For more information on object properties, refer to "Managing Form Objects" on page 302 and to chapter 6, "Working with Fields and Active Objects" on page 365.

When you want to display or modify form or object properties, 4th Dimension lets you choose between two tools:

- For form properties, you can choose between using the Form Properties window and the Property List.
- For object properties, you can choose between the Object Properties window and the Property List.

For more information on those tools, refer to "Comparing the Property List and property windows" on page 286.

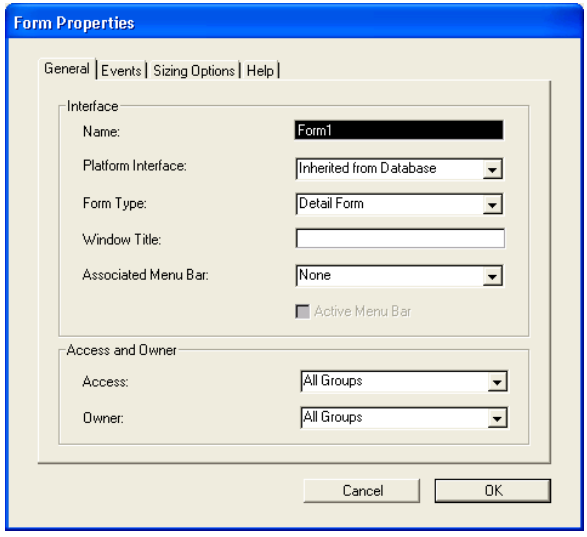
Form Properties

You can modify the current form's properties at any time. Modifications of the from properties can be done in both the Property List and the Form properties window.

▶ To display form properties in the form properties window:

1 **Choose Form Properties from the Form menu.**
OR
Choose Form Properties from the contextual menu that appears when you right-click (Windows) an empty area or press the Ctrl key while clicking an empty area (MacOS).

The Form Properties window consists of four pages, each of which can be selected using tab controls.



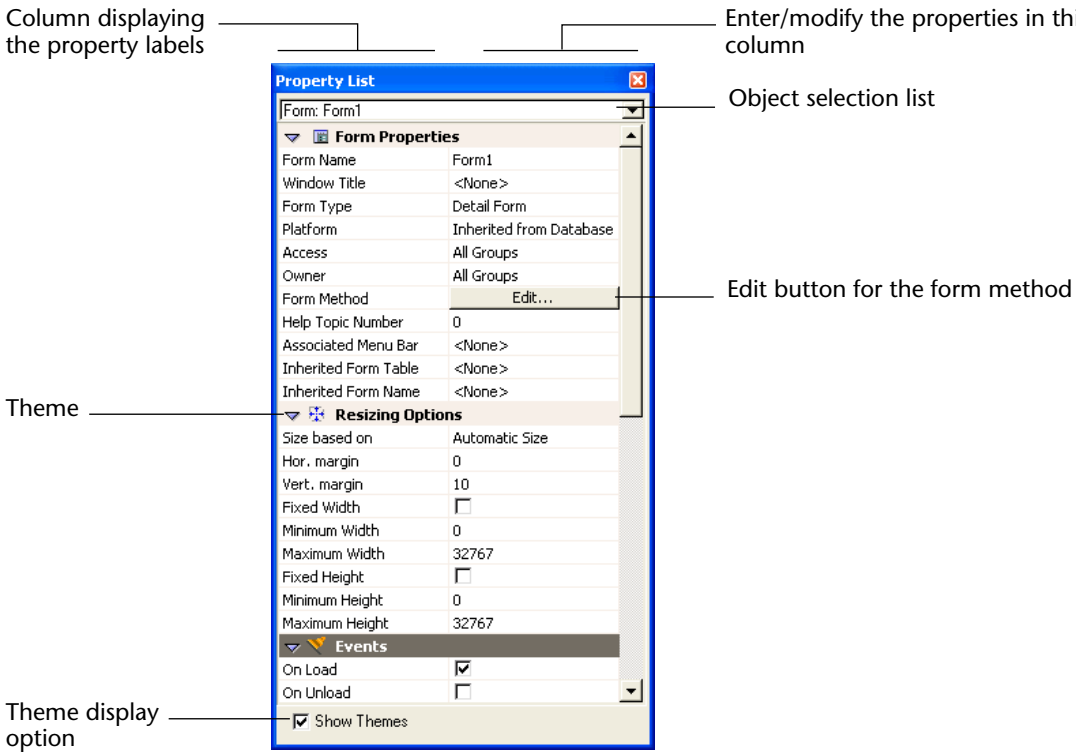
▶ To display the form properties in the Property List:

1 **Select Property List from the Display submenu in the Form menu.**
OR
Select Property list from the Display submenu of the contextual menu.

Note To toggle the display of the Property List when it is already selected in the **Display** submenu, press **Command+Shift+Space Bar** (MacOS) or **Ctrl+Shift+Space Bar** (Windows).

2 **Click an empty area of the form.**
OR
Select "Form: *FormName*" from the object list located at the top of the Property List.

The Property List displays the current form's properties:



Note You can expand or collapse each theme by clicking the triangle located on the left of the theme name. It allows you to expand only the themes in which you want to work. You can also choose not to display the theme labels by deselecting the **Show themes** check box. If you deselect this option, properties will be displayed in alphabetical order.

Object Properties

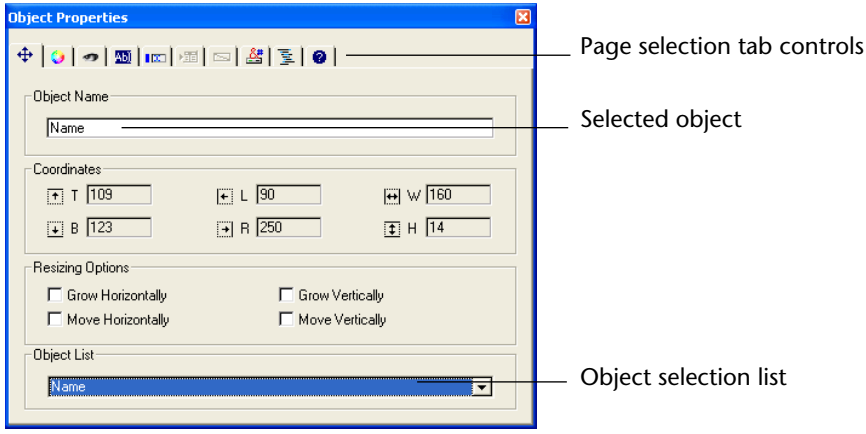
When a form is opened in the Form editor, you can display and modify the properties of any object. You can modify object properties using either the Property List or the Object Properties window.

► To display the object properties in the Object Properties window:

1 **Select Object Properties from the Display submenu of the Form menu.**
OR
Select Form Properties from the Display submenu of the contextual menu that appears when you right-click (Windows) the object or press the Ctrl key while clicking the object (MacOS).

Note To toggle the display of the Object Properties Window when it is already selected in the **Display** submenu, press **Command+Shift+Space Bar** (MacOS) or **Ctrl+Shift+Space Bar** (Windows).

2 **Click the object whose properties you want to display.**
OR
Select the object's name from the Object Selection list located at the bottom of the window.



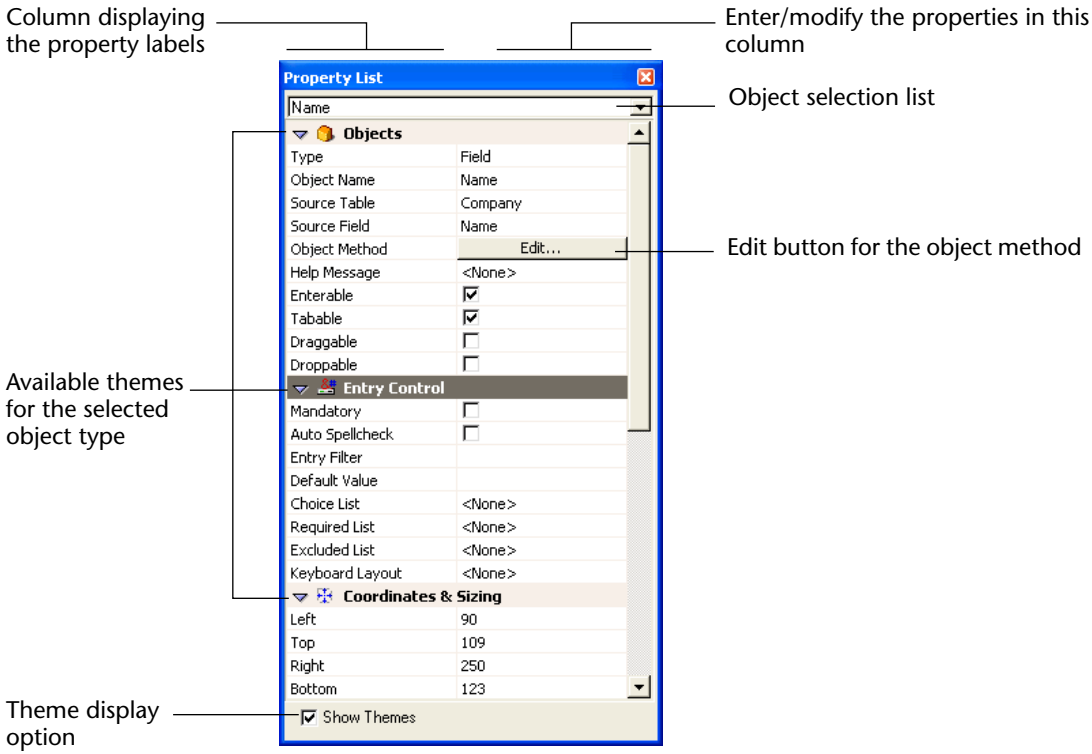
► To display object properties in the Property List:

1 **Choose Property List from the Display submenu in the Form menu.**
OR
Choose Property List from the Display submenu of the form contextual menu.

Note To toggle the display of the Property List when it is already selected in the **Display** submenu, press **Command+Shift+Space Bar** (MacOS) or **Ctrl+Shift+Space Bar** (Windows).

2 **Click the object in the form.**
OR
Select the object name from the Object Selection List located at the top of the Property List.

The Property List displays the current object's properties:



Comparing the Property List and property windows

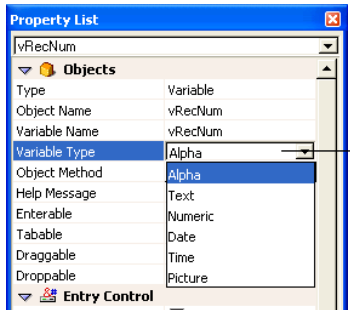
The theme logic in the Property List is different from the logic used in the Object/Form Properties window (tab controls in the latter do not match the themes in the Property List).

In addition, although both types of windows offer the same type of features, the Property List differs from the Object/Form properties window in the following ways.

- The contents of the Property List depend on the object type or form that is selected. In other words, themes and properties are appropriate for the currently selected object.

  In addition to the selective display of properties based on the object type, the Property List is dynamically updated when you modify an object's or a form's properties. For example if you set the Enterable property for an object, the Tabable property is displayed and becomes selectable.

Finally, the Property List allows you to set the type of the form variables. This allows you to modify the defaults.



Selecting the variable type

- As is described above, the Property List displays the properties of either the currently selected object or the current form. This allows you to switch between the form and object properties without having to switch tools.
  In addition, the Property List provides direct access to the form method. Keep in mind that the definition of an inherited form[1] and the positioning of the tabs[2] is carried out only through the Property List.
- The Property List manages objects within "families." This allows you to change an object's type and still maintain coherent data.
- The Property List does not allow you to edit help messages. This function is only available in the Object Properties window.
- To quickly navigate in the Property List, you can use the following keyboard shortcuts:
  - **Arrow keys** Used to go from one cell to another.
  - **PgUp** and **PgDn** Used to select the first or last visible cell of the property list.
  - **Home** and **End** Used to select the first or last cell of the property list.
  - **Ctrl + click** (Windows) or **Command + Click** (MacOS) on a theme label Used to Collapse/Expand every theme in the list.
  - **Ctrl + click** (Windows) or **Command + Click** (MacOS) on an event Used to select/Deselect every event in the list.

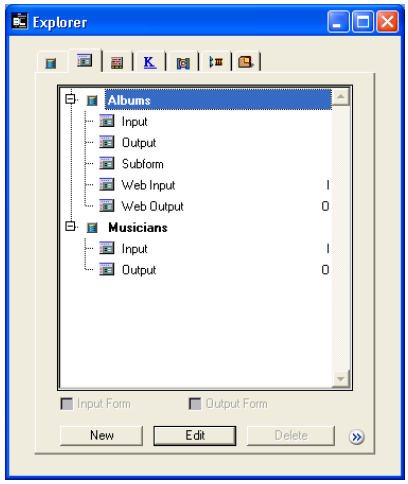
1. For more information on inherited forms, please refer to the paragraph "Inherited Forms," page 355
2. For more information about the positioning of the tabs, refer to the paragraph "Tab Controls", page 445.

Opening a Form in the Form Editor

You can edit an existing form in the Form editor at any time.

► To open a form in the Form editor:

1 **Choose Edit Form from the Design menu.**
OR
Hold down the Ctrl key (Windows) or the Command key (MacOS) and double-click the table title whose forms you want to open in the Structure editor.
OR
In the Structure editor, click the table name using the right mouse button (Windows) or press the Control key while clicking the table name (MacOS), then select Show Table Forms from the contextual menu.
4th Dimension displays the Forms page of the Explorer. You can expand any of the table names to display the forms associated with it.



If you have double-clicked a table's name or used the contextual menu of the Structure editor, that table is already selected.

2 **If necessary, expand the table name that contains the form you want to modify.**

3 **Select the name of the form you want to modify and click the Edit button.**
OR
Double-click the name of the form.
4th Dimension displays the form in a Form editor window.

4D Server Object locking occurs when two or more users attempt to modify the same form simultaneously. If a user is modifying a form in the Design environment, the form is locked. Other users cannot modify that same form until the first user frees the form by closing it.

Setting Form Properties

Form properties can be set in the Property list or in the From properties window. For more information on those tools, refer to "Displaying and Setting Form and Object Properties" on page 282.

Naming the Form

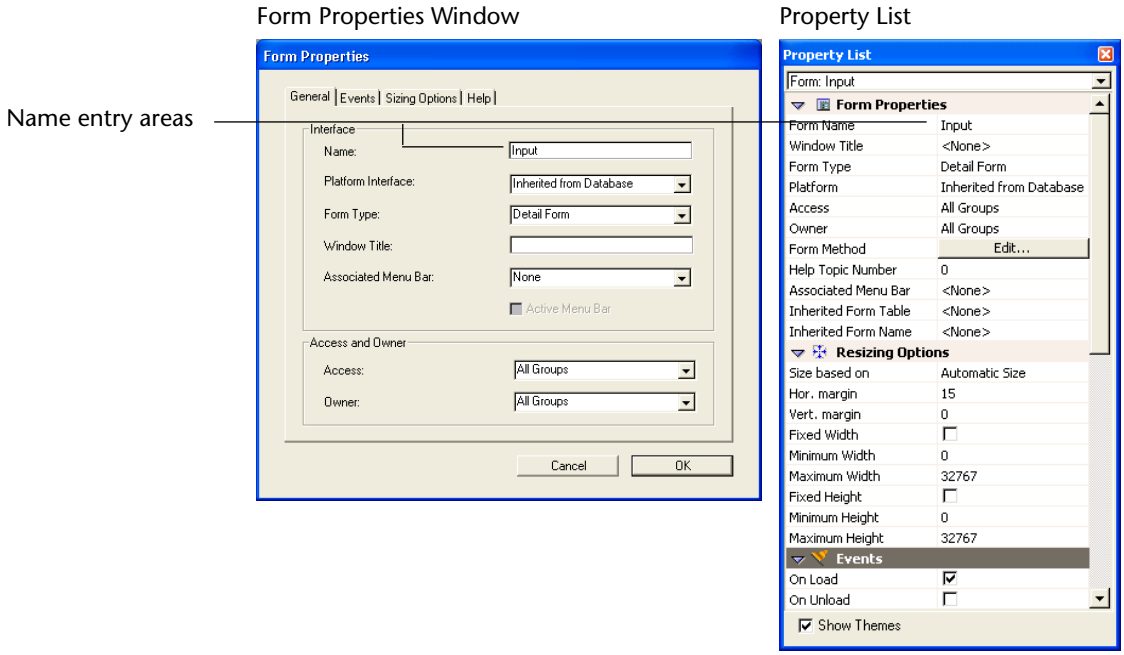
You can give the form a name in the Form Properties window. You can also rename a form using the Explorer. You use the names of forms when you are establishing default input and output forms for a table and in commands that accept a form name as a parameter, such as INPUT FORM and OUTPUT FORM.

You cannot use the same name for more than one form per table. This will confuse 4th Dimension when you try to refer to a form by name. You can, however, use the same name in different tables. For example, you can name all your input forms "Input" and all your output forms "Output."

▶ To name a form using the Form Properties window:

1 **With the desired form in the frontmost window, choose Form Properties or Display>Property List from the Form menu.**

These two choices are described in "Displaying and Setting Form and Object Properties" on page 282.



2 **Enter a name in the Name area.**

3 **If you have used the Form properties window, click OK to put it away.**

► To rename a form using the Explorer:

1 **Click the Forms tab to display the Forms page.**
A hierarchical list of tables and forms appears.

2 **Expand the table containing the form you would like to rename.**

3 **Hold down the Ctrl key (Windows) or the Command key (MacOS) and click on the form name.**
The form name becomes editable.

4 **Replace the current form name with the new name.**

5 **Press Tab or click anywhere outside the entry area to save the new name.**

If you rename a form that is referred to elsewhere in the database (such as in methods), update the references to the form.

Setting Form Access

You can control access to a form by setting Access and Owner privileges for groups of users. A single group can be assigned for each privilege using the Access and Owner drop-down lists. For information about creating a password access system with users and groups, see chapter 10.

The Access drop-down list controls which group can use the form in the User environment or in custom applications. If a user that is not in this group attempts use the form, 4th Dimension displays a message saying that the user's password does not allow him or her use the form.

The Owner drop-down list controls which group can edit the form in the Design environment. If a user who is not in this group attempts to edit the form in the Design environment, 4th Dimension displays a message saying that the user does not have the access privilege to edit the form.

Users who are assigned to both groups can use the form in both the User and Design environments and in custom applications.

▶ To set access privileges for a form:

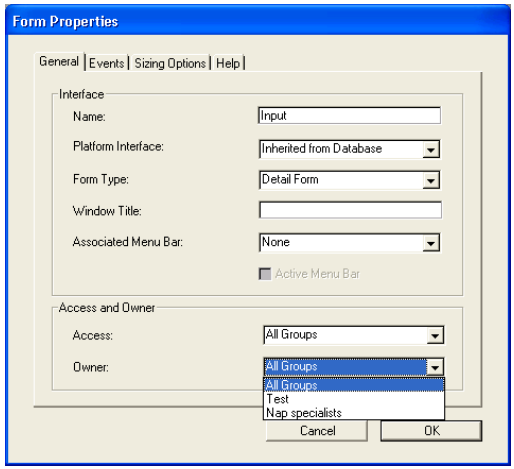
1 **Display the form properties in the Form Properties window or in the Property List.**

These two choices are described in "Displaying and Setting Form and Object Properties" on page 282.

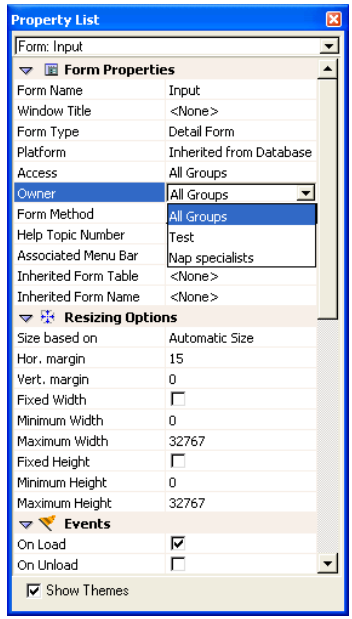
2 **Use the Access and Owner drop-down lists to make the desired access privileges assignments.**

The names of existing groups are displayed in each drop-down list.

Form Properties Window



Property List



3 **If you have used the Form properties window, click OK to put away the Form Properties window.**

Setting the Platform

You can set the platform interface for the entire database (in the Preferences dialog box) and for individual forms in the Form Properties window. At the form level, your choices are:

- **Inherited from Database** Use the platform interface selection you made in the Preferences dialog box.
- **Automatic** Use the platform interface defined in the MacOS and Windows menus in the User Interface page in the Preferences dialog box. Fore more information, please refer to the paragraph “Look Page,” page 110.
- **Mac OS 7**, **Windows 3.11, NT 3.51**, **Windows 95/98/2000, NT 4**, **Mac OS 9** or **Mac Theme** Choose a specific platform interface that will be used regardless of which platform on which the database is actually running.

  The default platform interface setting for a form is **Inherited From Database**.

► To set a platform interface for a form:

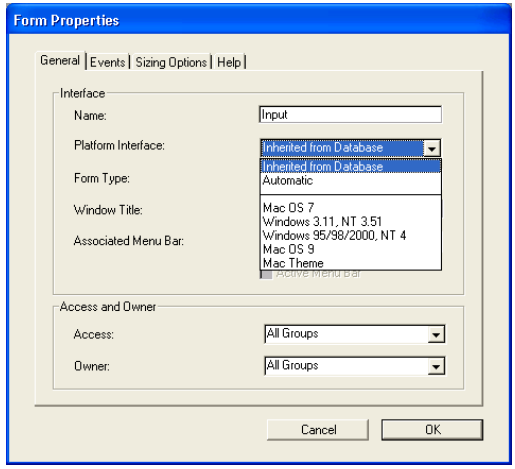
1 **Display the form properties in the Form Properties window or in the Property List.**

These two choices are described in "Displaying and Setting Form and Object Properties" on page 282.

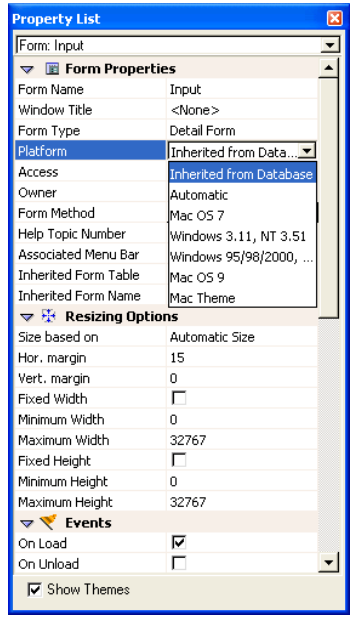
2 **Choose the desired platform interface from the Platform Interface drop-down list.**

Names of platforms are displayed in each drop-down list.

Form Properties Window



Property List



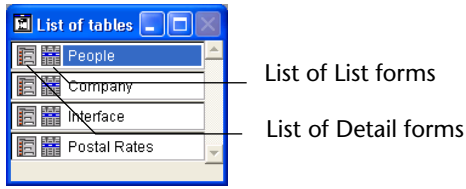
3 **If you have used the Form properties window, click OK to put it away.**

Choosing a Form Type

You can set the form type (detail form, list form, detail form for printing, or list form for printing) for a specific form.

This property allows you to restrict the number of forms displayed in the current Input and Output form selection lists (the **List of tables** window in the User environment).

Only forms whose type corresponds to the list are displayed.



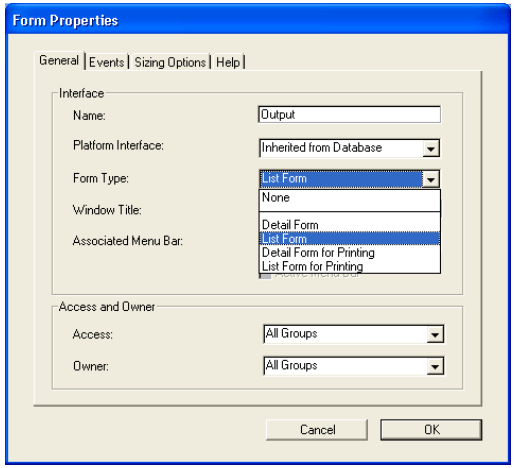
► To modify the form type:

1 **Display the form properties in the Form Properties window or in the Property List.**

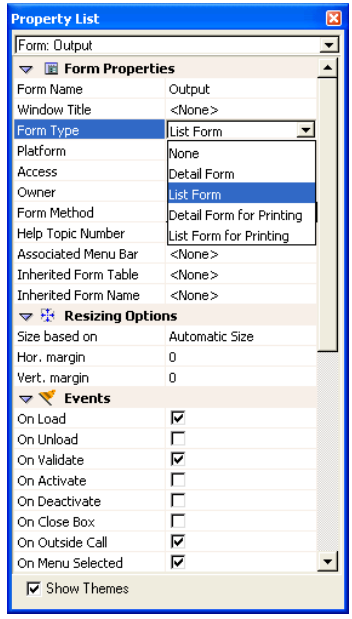
These two choices are described in "Displaying and Setting Form and Object Properties" on page 282.

2 **Select the desired form type, from the Form Type drop-down list.**

Form Properties Window



Property List



3 **If you have used the Form properties window, click OK.**

When the form type is "None," it is displayed in both menus of the List of Tables window.

Setting the Default Window Title

The default window title is used when the form is opened using the Open window and Open form window functions in custom applications. The default window title appears in the Title bar of the window. To set the default window title, enter it in the Window Title entry area.

You can use dynamic references to define the window title. The reference is resolved at the time the INPUT FORM command is called (when the * parameter is passed and the call to that command follows a call to Open window) and Open form window.

The following types of dynamic references can be inserted in a window's title:

- a **STR# resource reference**: the syntax to apply is ":16000,2" where 16000 is the resource number and 2 is its element.
- a **table or field label**: the syntax to apply is <?[TableNum]FieldNum> or <?[TableName]FieldName>. For more information, refer to "Inserting Dynamic Table and Field Names" on page 370.
- a **variable or a field**: the syntax to apply is <VariableName> or <[TableName]FieldName. The current value of the field or variable will be displayed in the window title.

Note The number of characters for a window title is limited to 31 characters.

Assigning a Menu Bar to a Form

When you create a custom database, you create custom menus. Custom menus allow you to add menu commands for automating specific tasks in the database, such as, for example, creating a report.

Custom menus are created in the Menu Bar editor. Each menu bar that you create includes at least one menu and is assigned a unique ID number and name. For more information on creating menu bars, menus and menu commands, refer to chapter 9, "Creating Custom Menus" on page 589.

When in User environment, a menu bar that is assigned to a form is added to the right of the current environment menu bar. A form menu bar is added only when the form is used for data entry.

In custom menus, a menu bar that is assigned to a form is added to the right of the current menu bar. The custom menu bar menu bar disables other menus unless the **Active Menu Bar** option is selected. If the custom menu bar is identical to the current menu bar, it is not added.

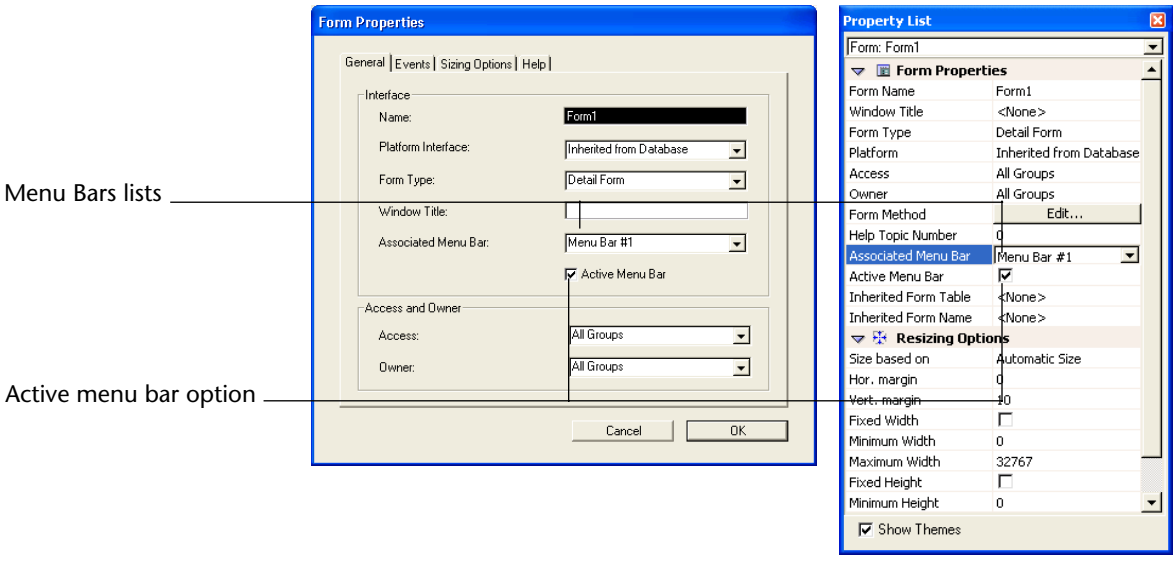
► To assign a menu bar to a form:

1 **Display the form properties in the Form Properties window or in the Property List.**

These two choices are described in "Displaying and Setting Form and Object Properties" on page 282.

Form Properties Window

Property List



2 **Select a menu bar from the "Associated menu bar" List.**

When the current form is displayed in User environment, the selected menu bar will be displayed to the right of the current menu bar.

3 **If you want to use that form in a custom application, select the "Active Menu Bar" option.**

This option tells 4th Dimension not to disable the current menu bar. If this option is not selected, 4th Dimension disables the current menu bar and only permits access to the form's menu bar.

4 **If you have used the Form Properties window, click OK to put it away.**

Settinizing and Resizing Options

When an input form is displayed in a custom application, you ordinarily open the form using the Open window or Open form window functions. Open window lets you specify the top, left, bottom, and right coordinates of the window as well as the window type. If you do not use any resizing options, the user's ability to resize the window depends on the window type.

Open form window creates a new window using the sizing and resizing options that are set in the Form editor.

The sizing and resizing options available in the Form Properties window give you greater control over sizing and resizing of the window.

Size Options

Size Options gives you control over the initial size of the window. Your choices are:

- **Automatic size** The size calculated by 4th Dimension based on the size necessary to display all objects on the form.
- **Set size** The size you enter in the width and height entry areas.
- **Size based on selected form object** The size based on a selected form object. 4th Dimension uses the smallest size necessary to display the selected object. For example, if you choose an object that is placed at the bottom-right corner of the area to be displayed, 4th Dimension will open a window just large enough to include the object.

  Choose this option if you need to place some active objects in an off-screen area (i.e., outside the bounding rectangle). When you use this option, these objects will not affect the size of the window.

  When you select either automatic size or size based on a selected object, the Width and Height areas change to Hor. Margin and Vert. Margin, respectively. You can then enter values (in pixels) used to set additional margins that are added to the edges of the form window. These values also determine the top and right-hand margins of the forms used in the Label editor (see the paragraph "Printing Labels", page 512).

► To set the sizing options:

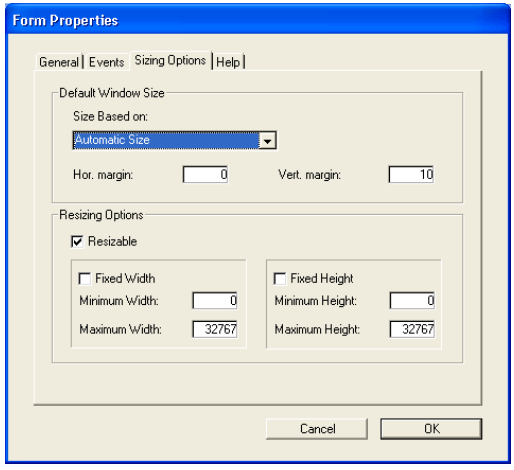
1 **Display the form properties in the Form Properties window or in the Property List.**

These two choices are described in "Displaying and Setting Form and Object Properties" on page 282.

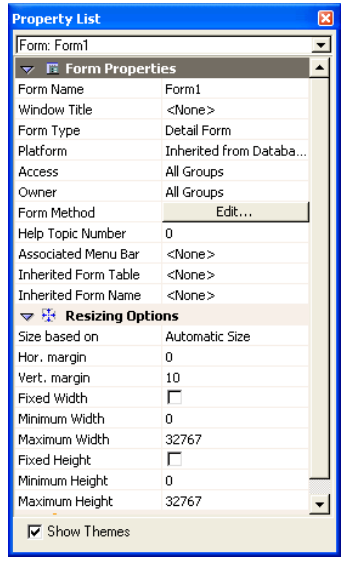
2 **In the Form Properties Window, click the Sizing Options tab.**
OR
In the Property List, expand the "Resizing Options" theme.

The Sizing options are displayed.

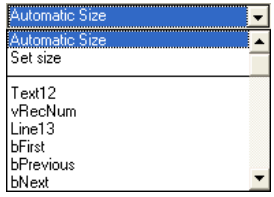
Form Properties Window



Property List



In the Default Window Size area, the Size Based On drop-down list controls the initial size of the window.



Names of objects on the current form

The two options above the dividing line let you choose either Automatic or Set Size. If you choose Set Size, you must enter the desired Width and Height into the appropriate entry areas.

All the objects on the current form are listed below the dividing line in the drop-down list. To size the window based on a form object, choose the desired form object from the drop-down list.

3 **Choose the desired sizing option from the drop-down list.**

If you did not choose Set Size, the Width and Height areas change to Horizontal Margin and Vertical margin.

4 **If you chose Set Size, enter the desired width and height in the entry areas.**
OR
If you chose either Automatic Size or a form object, enter the desired horizontal and vertical margins (in pixels) in the appropriate entry areas.

Resizing Options

The resizing options in Form Properties let you make windows of any type resizable in custom applications and let you set the minimum and maximum sizes. Setting the minimum size is a way of preventing users from resizing a form so that the button panel or other essential objects are no longer visible.

► To make the window resizable:

1 **Click the Resizable check box.**

When you click Resizable, the Fixed Width and Fixed Height check-boxes are enabled and the minimum and maximum entry areas become enterable (in the Property List, those areas are displayed only if the Resizable option is selected).

2 **Click either Fixed Width or Fixed Height to prevent resizing in either direction (Optional).**

For example, you may have a column of buttons and you want to prevent the user from resizing the window so that some of the buttons are no longer visible. You would choose Fixed Height.

3 **Enter a minimum value in either dimension to prevent the user from making the window too small (Optional).**
OR
Enter a maximum value to prevent the user from making the window too big (Optional).

You would use this option to prevent the user from hiding necessary entry areas or controls.

4 **If you have used the Form properties window, click OK to put it away.**

Form Events

You can write a form method using the Method editor. If you do so, you should indicate which form events should be executed. You do this in the Events page of the Form Properties window or in the Events theme of the Property List. When the form is used, only the events that you select will actually occur.

For information on the available events, see the section "Form and Object Events" on page 525.

Your database will run faster if you deselect superfluous events.

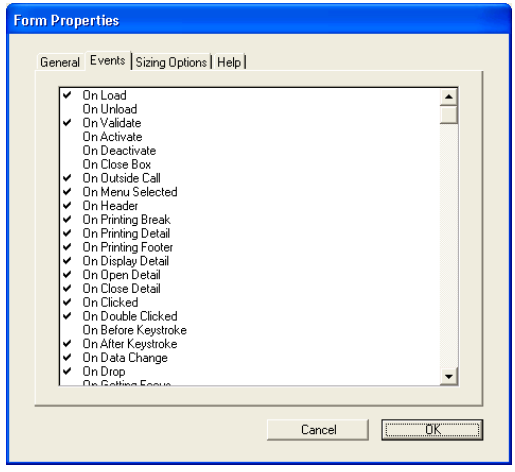
► To activate events for the form:

1 **Display the form properties in the Form Properties window or in the Property List.**

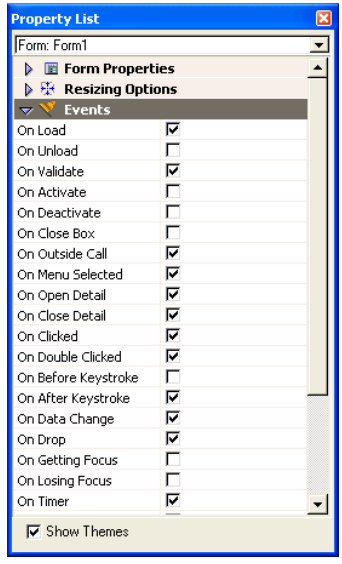
These two choices are described in "Displaying and Setting Form and Object Properties" on page 282.

2 **In the Form Properties Window click the Events tab.**
OR
In the Property List, expand the "Events" theme.
The list of events is then displayed:

Form Properties Window



Property List



A check box next to the event indicates that the event will occur when the form is used.

3 **Select only the events that are needed.**

To select or deselect all events, hold down **Ctrl** (**Command** under MacOS) and click an event.

4 **When you are finished setting Form properties, click OK.**

Contextual On-line Help

4th Dimension allows you to associate a custom on-line help file to each database. The creation of help files is described in Appendix B on page 757.

Help files can be contextual, which means that they can display information related to the context from which they were called. To do so, you can associate a precise section of this help file with each of the database's forms.

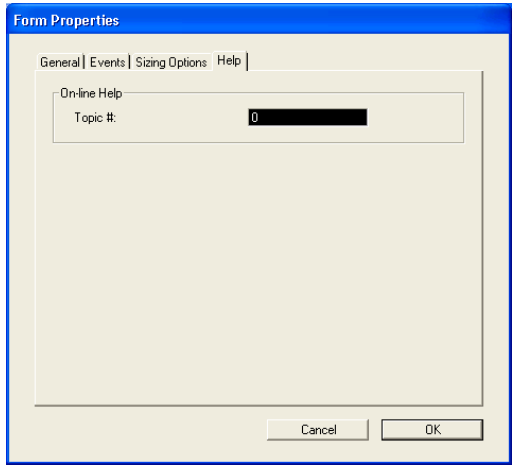
► To associate a section number with a form:

1 **Display the form properties in the Form Properties window or in the Property List.**

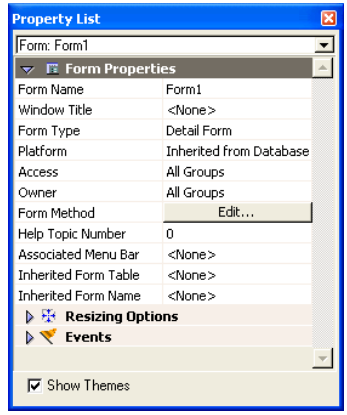
These two choices are described in "Displaying and Setting Form and Object Properties" on page 282.

2 **In the Form Properties window, click the Help tab.**
OR
In the Property List, expand the "Form Properties" theme.

Form Properties Window



Property List



3 **Enter the help section number you want to associate with the form in the Help Topic Number area.**

4 **If you have used the Form properties window, click OK.**

You can repeat this procedure for each form that needs to be assigned a help file.

Make sure you assign help topic numbers that match numbers defined in the help file. For more information on this point, refer to Appendix B on page 757.

Managing Form Objects

You customize a form by creating and manipulating objects in the form. You use the Form editor to draw the objects, modify them, arrange them, set their properties, or delete them.
You can select and modify any object in a form, including the fields, buttons, and graphic objects created by the Form Wizard.

Selecting Objects

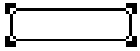
Before you can perform any operation on an object (such as changing a line width or font), you need to select the object that you want to modify.

When the Form Wizard creates a form, it creates one or more display pages and a background page. The Form Wizard may place buttons, the form title, and decorative rectangles on the background page. Objects on the background page are selectable only from the background page. If you have difficulty selecting an object that was created by the Form Wizard, switch to the background page and try again. You can also choose to hide any element located on the background page by deselecting **Page 0** from the **Display** submenu of the **Form** menu (or the Form editor contextual menu).

► To select an object using the Tools palette:

1 **Click the Arrow tool in the Tools palette.**
When you move the pointer into the form area, it becomes a standard arrow-shaped pointer.

2 **Click the object you want to select.**
Resizing handles identify the selected object.



► To select an object using the Object Properties window or the Property List:

■ **Choose the object's name from the Object List drop-down list located at the bottom of the Object Properties window.**
OR
Choose the object's name from the Object List drop-down list located at the top of the Property List.
Using these two methods, you can select an object that is hidden by other objects or located outside the visible area of the current window.

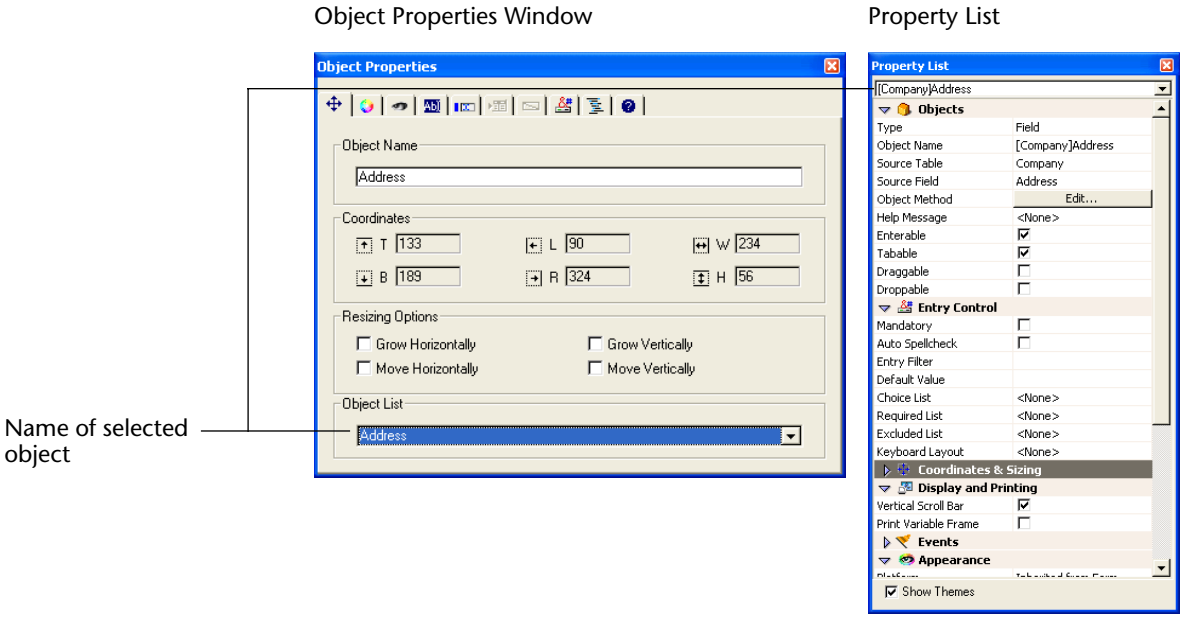
To deselect an object, click outside the object's boundary or **Shift+click** the object.

Note It is also possible to select objects by double-clicking them in the result window of an overall search in the database. For more information on this point, refer to the paragraph "Searching in the Database," page 94.

Viewing or Modifying Object Properties

You view or modify an object's properties using the Object Properties window or the Property List (for more information about these two methods, refer to "Displaying and Setting Form and Object Properties" on page 282). To select the window you want to work with, select it from the **Display** submenu of either the **Form** menu or the Form editor contextual menu. To hide or show the selected window, double-click the object or press **Ctrl+Shift+Space** (**Command–Shift–Space** under MacOS).

When either window is on screen, click the object or select it from the Object List drop-down list.



The Object List drop-down list contains the names of all the objects on the current page. You can select an object and view its properties by choosing its name from this list.

Viewing or Modifying an Object's Method

To view the object's method or create a new method for the object hold down the **Alt** key (Windows) or **Option** key (MacOS) and click the object[1]. For more information about object methods, see the section "Using Object Methods with Fields and Objects" on page 462 and "Object Methods" on page 518.

Selecting Multiple Objects

You may want to perform the same operation on more than one form object — for example, to move the objects, align them, or change their appearance. 4th Dimension lets you select several objects at the same time. There are four ways to select multiple objects:

- Choose **Select All** from the **Edit** menu to select all the objects.
- Hold down the **Shift** key and click the objects you want to select.

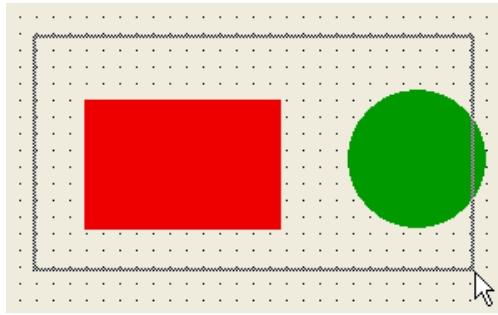
1. Only active objects can have methods, but all objects have properties. If an object cannot have a method, the Events tab in the Object Properties window is dimmed and the Property List contains no Object Method line.

- Start at a location outside the group of objects you want to select and drag a marquee (sometimes called a selection rectangle) around the objects.
- Hold down the **Alt** key (Windows) or the **Option** key (MacOS) and draw a marquee. Any object that is completely enclosed by the marquee is selected.

  You draw a marquee with the arrow pointer. A marquee defines a rectangular region that select objects it surrounds or touches.

  To select objects by drawing a marquee around them, you must press the mouse button down and start dragging *in an area that contains no objects*. When you release the mouse button, if any part of an object lies within the boundaries of the selection rectangle, that object is selected.

  The figure below shows a marquee being drawn to select two objects:



  To deselect an object that is part of a set of selected objects, hold down the **Shift** key and click the object. The other objects remain selected. To deselect all the selected objects, click outside the boundaries of all the objects.

Moving Objects

You can move any graphic or active object in the form including fields and objects created with a template.

When moving an object, you have the following options:

- Move the object by dragging it.
- Move the object one pixel at a time using the arrow keys.
- Move the object in the user-defined increments established in the Grid Definition dialog box.

- Use the Coordinates page of the Object Properties window (described in the section "Resizing Objects" on page 307).

▶ To move an object by dragging:

1 **Select the object or multiple objects that you want to move.**

2 **Move the pointer over the selected object or one of the objects in a selected set of objects and drag to the new location.**

4th Dimension displays markers that show the location of the object's boundaries in the rulers so that you can place the object exactly where you want it. As you begin dragging the selected object, its handles disappear.

Be careful not to drag a handle. Dragging a handle resizes the object.

3 **Release the mouse button to complete the move.**

▶ To move an object one pixel at a time:

1 **Select the object or objects you want to move.**

2 **Use the arrow keys on the keyboard to move the object.**

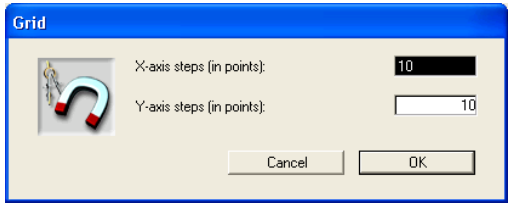
Each time you press an arrow key, the object moves one pixel in the direction of the arrow.

You can also use the arrow keys to move an object x pixels at a time, when x is a value set in the Define Grid dialog box.

▶ To move an object in user-defined increments:

1 **Choose Grid Definition from the Form menu.**

The Grid Definition dialog box appears.



2 **Enter the desired values in the X-axis and/or Y-axis entry areas and click OK.**

3 **Choose Turn Grid On from the Form menu or from the Form editor contextual menu.**

4 **Hold down the Shift key and use the arrow keys to move the object in the increments you specified.**

Resizing Objects

You can change the size of any object that appears on the form. 4th Dimension lets you stretch or shrink objects on the form.

When resizing objects, you have the following four options:

- Resize an object by dragging a resizing handle.
- Resize an object one pixel at a time by using the **Ctrl** key (Windows) or the **Command** key (MacOS) and the arrow keys.
- Resize an object in user-defined increments using settings established in the Grid Definition dialog box.

► To resize an object by dragging:

1 **Select the object you want to resize.**

2 **Move the pointer over one of the four handles that appear on the selected object.**

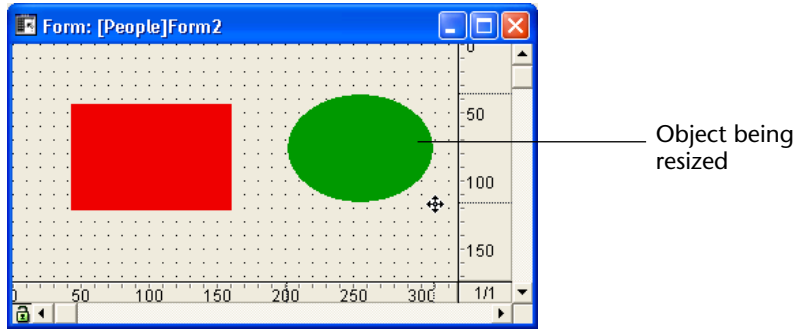
The pointer changes into a multi-directional arrow ✣ and the handles disappear.

3 **Drag the handle toward the center of the object to shrink it.**
OR
Drag the handle away from the object's center to enlarge it.

4th Dimension resizes the object.

As you drag the handle, the corner of the object opposite the dragging handle remains stationary.

The following illustration shows an object being resized.



Note If you press **Shift** and then drag the handle, the movement is constrained. Lines can then be only vertical, 45°, or horizontal, rectangles can be only square, and ovals can be only circular.

► To resize an object one pixel at a time:

1 **Select the object you want to resize.**

2 **Hold down the Ctrl key (Windows) or Command key (MacOS) and use the arrow keys to resize the object.**

Pressing the up or down arrow keys resizes the object's height while pressing the left or right arrow keys resizes the object's width.

► To resize an object in user-defined increments:

1 **Choose Grid Definition from the Form menu.**

The Grid Definition dialog box appears.

2 **Enter the desired increments in the X-axis and/or Y-axis entry areas and click OK.**

3 **Choose Turn Grid On from the Form menu or from the Form editor contextual menu.**

4 **Select the objects you want to resize.**

5 **Hold down the Shift+Ctrl keys (Shift+Command key under MacOS) and use the mouse or the arrow key to resize the objects in user-defined increments.**

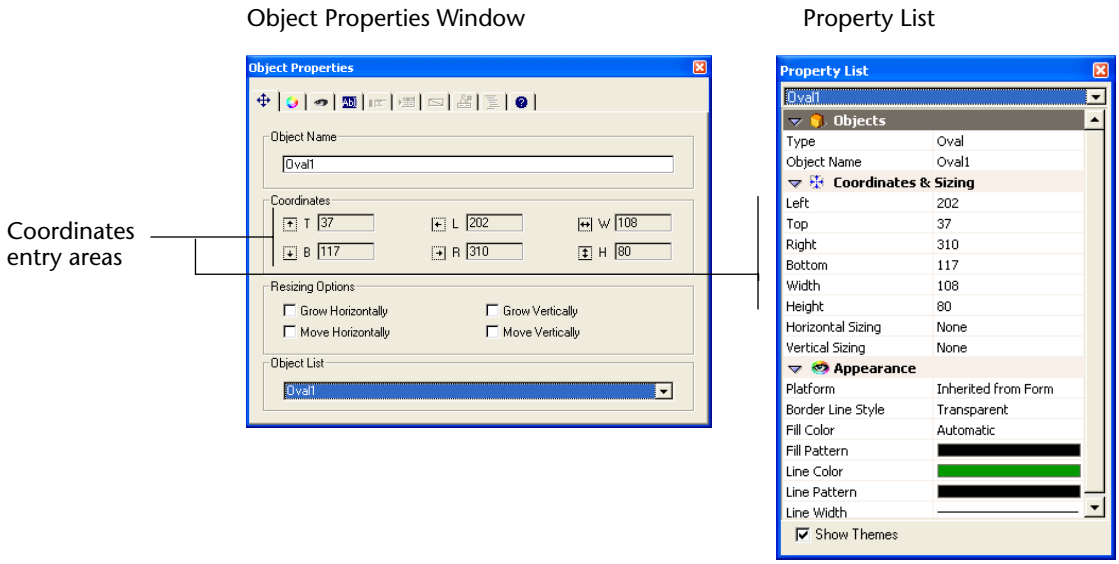
► To resize an object by entering coordinates:

1 **Display the Object Properties window or the Property List and click the object you want to resize.**
OR
Double-click the object.

2 **In the Object Properties window, click the Coordinates tab.**
OR
In the Property List, expand the "Coordinates & Sizing" theme.

The coordinates of the object appear. They are displayed in pixels, centimeters, or inches (depending on the ruler units).

The upper-left coordinates of the form area are 0,0.



3 Enter new values in the coordinate entry areas.

4th Dimension moves the boundaries of the object to the positions you entered. Depending on the values you use, the object may be resized or moved (or both).

Automatic Resizing and Repositioning

When the user resizes the window in which form is displayed, the objects that are included in that window can be resized or moved.

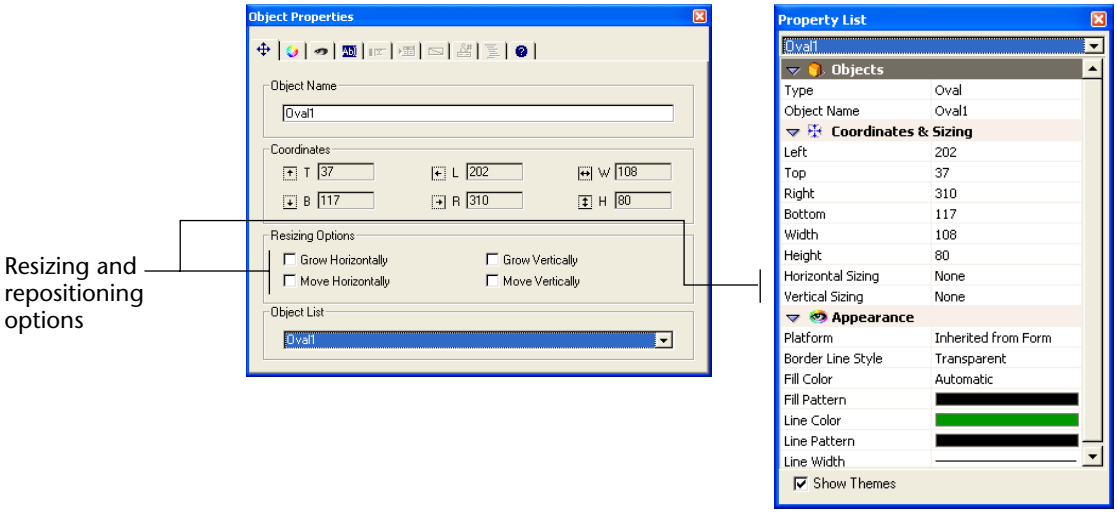
Automatic resizing works when a user resizes a window that displays a form. Automatic resizing causes an object to grow as the form is enlarged (or become smaller as the enlarged window is reduced). For example, if you use a rectangle that encloses the fields on an entry form, automatic resizing causes the rectangle to grow to the edges of the window as the user enlarges the window.

You can also enable automatic repositioning. Automatic repositioning moves an object either horizontally or vertically as the form is resized. When automatic repositioning is on, 4th Dimension tries to keep the object in view as the user reduces the size of the window. For example, if the user resizes a row of buttons so that some of the buttons become obscured, automatic repositioning tries to move the buttons either horizontally or vertically, so that they remain in view.

You enable automatic resizing or repositioning in the Coordinates page of the Object Properties window or in the Coordinates and Sizing theme of the Property List.

Object Properties Window

Property List



- In the Object Properties window, the resizing and positioning options are defined using four check boxes.
- In the Property List, there are two lines, **Horizontal sizing** and **Vertical sizing**, for which you can assign three properties (None, Grow, and Move).

Here is the effect of these options:

Object Properties Window	Property List Option	Result
Grow horizontally[1]	Horizontal Sizing: Grow	When the user resizes the width of the window, 4th Dimension applies the same percentage to the object's width.
Move horizontally	Horizontal Sizing: Move	When the user resizes the width of the window, 4th Dimension moves the object left or right the same amount as the width increase.
Grow vertically	Vertical Sizing: Grow	When the user resizes the height of the window, 4th Dimension applies the same percentage to the object's height.
Move vertically	Vertical Sizing: Move	When the user resizes the height of the window, 4th Dimension moves the object up or down the same amount as the height change.

1. The options Grow horizontally and Move horizontally are mutually exclusive. The options Grow vertically and Move vertically are mutually exclusive.

The repositioning options enable the object to move in the specified direction to try to remain visible.

Using the Rulers

The Form editor rulers extend along the height and width of the form. A dotted grid appears in the Form editor whenever you have rulers showing. This feature helps you position objects precisely.

You can hide the rulers to increase your working area in the Form editor window. You can display the rulers again when necessary. To hide or display rulers, choose **Rulers** from the **Display** submenu in the **Form** menu or the Form editor contextual menu.

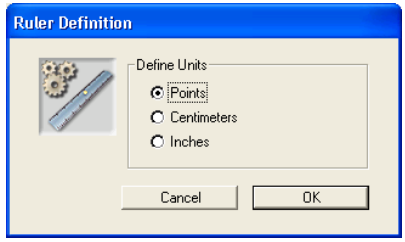
The rulers contain markers that show the position of the pointer when creating or resizing an object. While you are moving the object, the markers change to show the top and bottom and left and right sides of the object.

The object markers allow you to align other objects to the same position on the rulers.

You can change the units the rulers use to suit your preference.

► To define ruler units:

1 **Choose Define Ruler Units from the Form menu.**
4th Dimension displays the Ruler Definition dialog box, shown below.



2 **Click the measurement unit you want to use.**
- Click **Points** to display rulers that provide measurement in points. One point is equal to the width of one pixel. There are 72 points in an inch.
- Click **Centimeters** to display metric scale rulers.
- Click **Inches** to display rulers that use feet and inches.

3 **Click OK.**
4th Dimension changes the measurement units to the scale you have selected. The objects' coordinates will also use the same units. For information about the Object Coordinates dialog box, see the section "Resizing Objects" on page 307.

Creating Objects

You can use the Tools palette to add objects to a form. The object creation tools are located in the upper part of the Tools palette:

The object type is selected by clicking the appropriate tool in the Tools palette. Some buttons display a small triangle located in the lower right corner of the tool icon. This indicates that the corresponding object type includes several variations within that type. To display the variations for a type, click the tool's icon and hold down the mouse button until variations are displayed in a menu.

For a faster display, you can also right-click the tool's icon (Windows) or press the Ctrl key while clicking the tool's icon (MacOS). If you select a type from the menu, it becomes the current type and the tool's icon is updated to reflect that change.

For more information on the Tools palette and its tools, refer to "The Tools Palette" on page 273.

You can create any type of object (graphic or active) by drawing it or by dragging the tool from the Tools palette to the form.

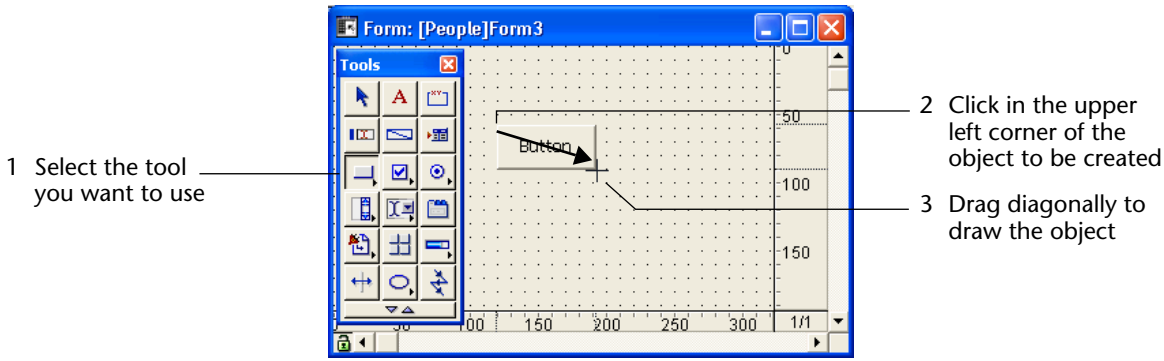
▶ To create an object by drawing:

1 Select the type or subtype of object you want to draw by clicking its tool in the palette.

The pointer becomes a crosshair when it is in the over the area in which you can draw the object. The pointer becomes an arrow when it is positioned over a menu, ruler, or the palette so that you can use it to select.

2 Drag to create an area for the object.

For two-dimensional objects (ovals, rectangles, fields, grids, text areas, active objects, and subform areas), drag diagonally.



Text areas include a text sample.

Note Hold down the **Shift** key as you draw to constrain the object to a regular shape. Lines are constrained to horizontal, 45°, or vertical, rectangles are constrained to squares, and ovals are constrained to circles.

3 When you have finished drawing the object, release the mouse button.

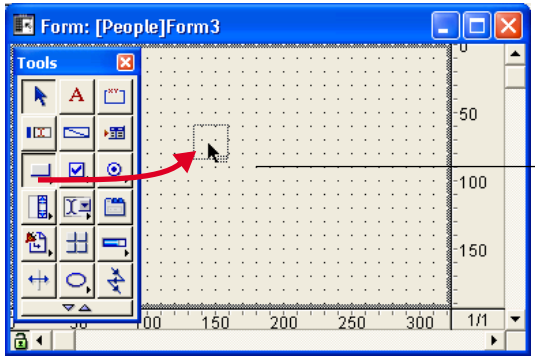
4th Dimension creates the object, assigns it a name and makes it the currently selected object. The Arrow tool is automatically selected and the pointer becomes an arrow, unless the last tool you have used is the Text tool.

If you created a field or active object, 4th Dimension also displays the Object Properties window or the Property List. You can use it to specify additional information.

► To create an object by drag and drop:

1 **If the desired object does not have a drop-down list associated with it, select the object by clicking its tool in the palette and drag it to the location on the form where you want the object to be.**
If the desired object is in a drop-down list, first select the object from the drop-down list and then drag the object from the palette to the form.
A dotted line indicates the location of the cursor.

2 **When you reach the location where you want the object to be, release the mouse button.**:



4th Dimension creates the object, assigns it a name, and makes it the currently selected object. The Arrow tool is automatically selected and the pointer becomes an arrow, unless the last tool you have used is the Text tool.

If you created a field or active object, 4th Dimension also displays the Object Properties window or the Property List. You can use it to define other object characteristics.

Grouping Objects

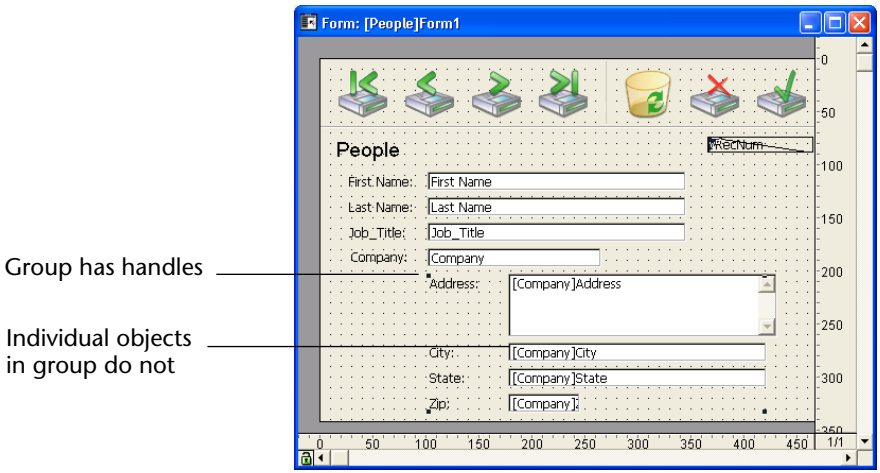
4th Dimension lets you group objects so that you can select, move, and modify the group as a single object.

Objects that are grouped retain their position in relation to each other. You would typically group a field and its label, an invisible button and its icon, and so forth.

When you resize a group, all the objects in the group are resized proportionally (except text areas, which are resized in steps according to their font sizes).

Grouping is also used for grouped scrollable areas. For more information, refer to the *4th Dimension Language Reference* manual.

Groups can be part of other groups. The following illustration shows grouped objects:



You can ungroup a group of objects to treat them as individual objects again.

An active object that has been grouped must be ungrouped before you can access its properties or method.

Grouping affects objects only in the Form editor. In the User environment, all grouped objects except for scrollable areas act as if they were ungrouped.

► To group objects:

1 **Select the objects that you want to group.**

2 **Choose Group from the Object menu.**
OR
Press Ctrl+G key (Windows) or Command–G (MacOS).
4th Dimension marks the boundary of the newly grouped objects with handles. No handles mark the boundary of any of the individual objects within the group.

Now, when you modify the grouped object, you change all the objects that make up the group.

► To ungroup an object:

1 **Select the grouped object that you want to ungroup.**

2 **Choose Ungroup from the Object menu.**
OR
Press Ctrl+H (Windows) or Command–H (MacOS).
If **Ungroup** is dimmed, this means that the object is already separated into its simplest form.

4th Dimension marks the boundaries of the individual objects with handles.

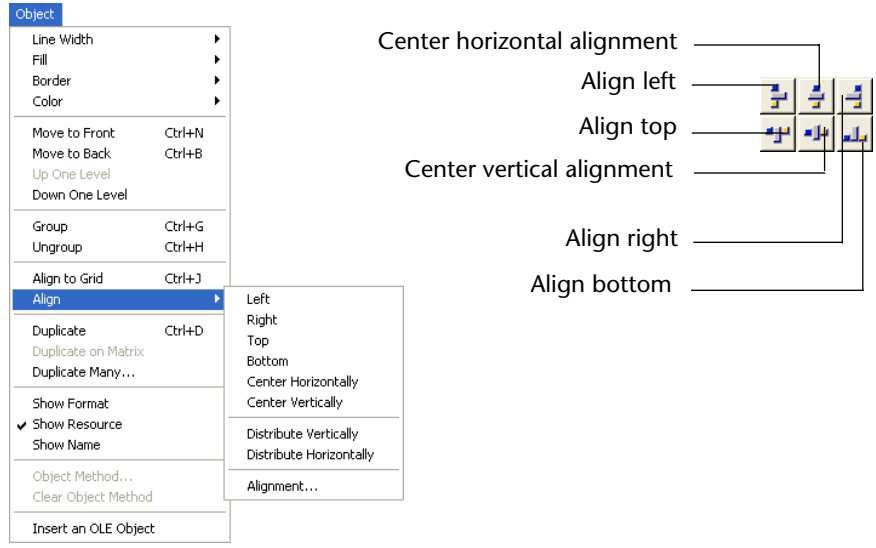
Aligning Objects

The Form editor's alignment tools and grid tools let you align objects to each other or to an invisible grid on the form.

- When you align one object to another, you can align it to the top, bottom, side, or horizontal or vertical center of the other object. You can directly align a selection of objects using the alignment tools or apply more advanced alignment settings using the Alignment Assistant. The latter option allows you, for example, to define the object will be used as the position reference and to preview the alignment in the form before applying it.
- When you use the invisible grid, you can align an object or collection of objects to a grid on the form.

Using the Instantaneous Alignment Tools

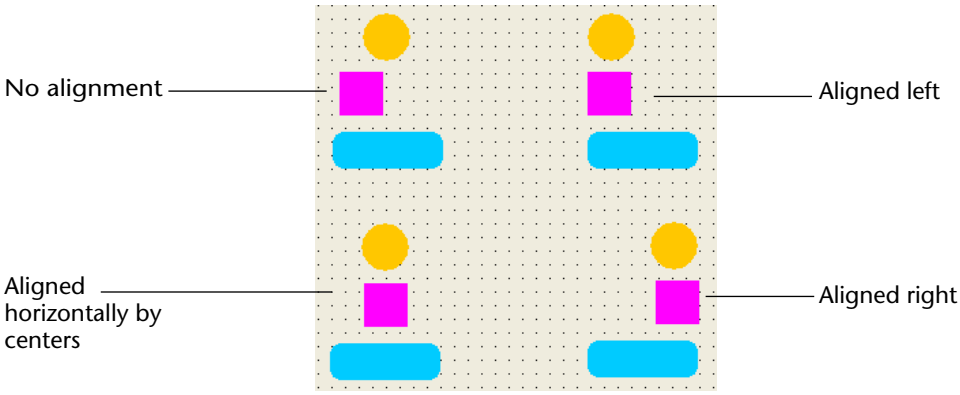
The alignment tools in the Tools palette and in the **Align** submenu of the **Object** menu allow you to quickly align selected objects.



The alignment icons in the Tools palette illustrate each alignment type.

When 4th Dimension aligns objects, it leaves one selected object in place and aligns the remaining objects to that one. This object is the "anchor." It uses the object that is the furthest in the alignment's direction as the anchor and aligns the other objects to that object. For instance, if you want to perform a right alignment on a set of objects, the rightmost object will be used as the anchor.

The figure below shows aligned objects:



► To align a set of objects:

1 **Select the objects you want to align.**
For more information on how to select objects, refer to "Selecting Multiple Objects" on page 304.

2 **Click the alignment tool in the Tools palette that corresponds to the alignment you want.**
OR
Choose an alignment menu command from the Align submenu in the Object menu.
OR
Choose an alignment menu command from the Align submenu in the contextual menu that is displayed when you right-click one object in the selection (Windows) or when you press the Control key while clicking an object in the selection.
4th Dimension aligns the selected objects according to the alignment you selected.

Using the Alignment Assistant

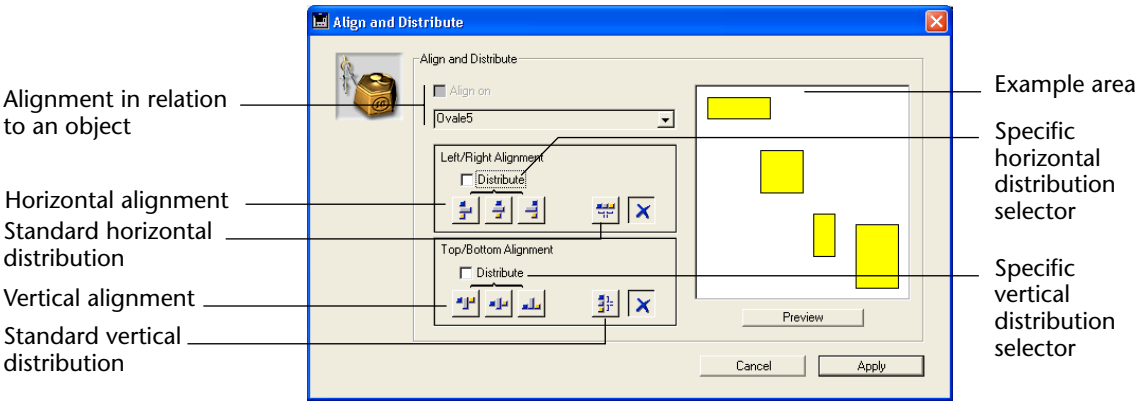
The Alignment Assistant allows you to perform any type of alignment and/or distribution of objects.

► To align objects using the Alignment Assistant:

1 **Select the objects you want to align.**
For more information on how to select objects, refer to "Selecting Multiple Objects" on page 304.

2 **Choose the Alignment command from the Align submenu in the Object menu.**
OR
Choose the Alignment command from the Align submenu in the Form editor contextual submenu.

The Alignment dialog box is displayed:



3 **In the "Left/Right alignment" and/or "Top/Bottom alignment" areas, click the Alignment icon that corresponds to the alignment you want to perform.**

The example area displays the results of your selection.

4 **To perform an alignment that uses the standard anchor scheme, click Preview or Apply.**

In this case 4th Dimension will perform an alignment using the standard anchor as described in "Using the Instantaneous Alignment Tools" on page 317.

OR

To align objects to a specific object, select the "Align on" option and select the object to which you want the other objects to be aligned from the object list.

In this case, the position of the reference object will not be altered.

You can preview the results of the alignment by clicking the Preview button. The objects are then aligned in the Form editor but since the Alignment dialog box does not go away, you can still cancel or apply the alignment.

Note The Alignment Assistant allows you to align and distribute objects in one operation. For more information on how to distribute objects, refer to "Distributing Objects" on page 322.

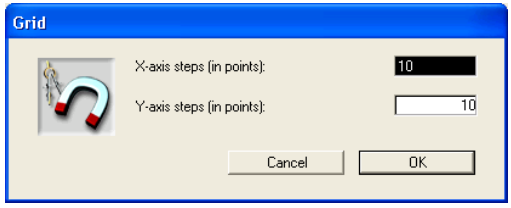
Using the Alignment Grid

The Form editor provides a grid of horizontal and vertical lines that help you place and align objects in a form. You can do the following:

- Specify the dimensions of the grid,
- Hide or show the grid,
- Turn the grid on (in this case, objects are automatically aligned to that grid when they are created or moved),
- Align a selection of objects to the grid, even if it is disabled.

▶ To define the grid:

1 **Choose Define Grid from the Form menu.**
4th Dimension displays the Grid Definition dialog box, shown below.



2 **Enter a number of points (72 points to an inch) for each step on the x-axis and y-axis.**
Grids are always defined in points. In effect, you are defining rectangles that are *x* points wide and *y* points high. For example, you might specify 10 points for x-axis steps and 20 points for y-axis steps.

If you want the x-axis and y-axis steps to be equal, enter identical point values in the boxes.

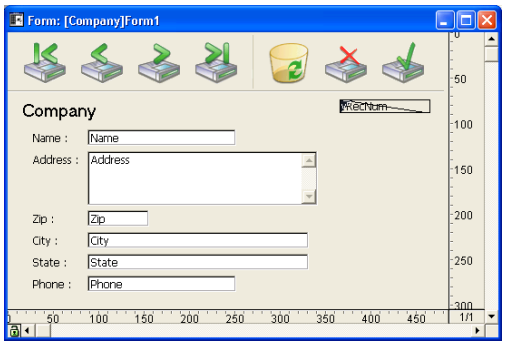
3 **Click the OK button to accept the grid settings.**
OR
Click the Cancel button to cancel the settings.

▶ To Hide/Display the grid:

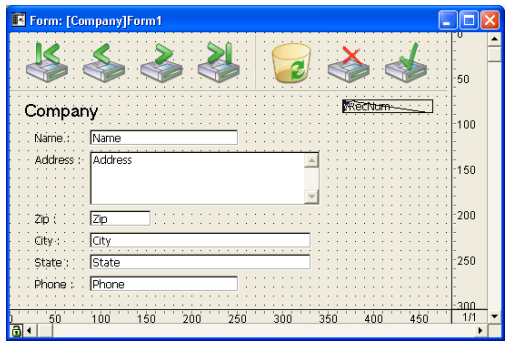
1 **Select Grid from the Display submenu in the Form menu or in the Form editor contextual menu.**

For more information, refer to "Showing/Hiding Elements in the Form Editor" on page 270.

Grid not displayed



Grid displayed



Note The display status of the grid is independent from its activation status. It can be displayed but inactive or vice-versa.

► To turn the grid on:

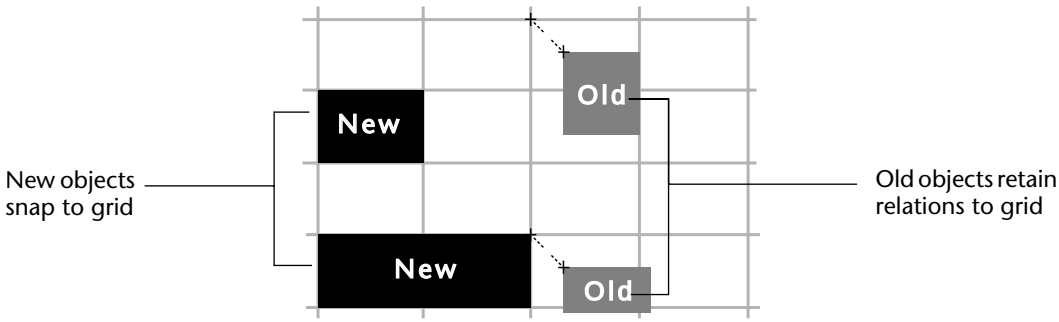
1 Choose Turn Grid On from the Form menu.

When the grid is on, there is a check mark next to the **Turn Grid On** menu item.

If you add an object to the form while the grid is on, it aligns to the invisible grid as you draw it. Each corner of the object's defining rectangle jumps to the nearest intersection of the grid.

Existing form objects are not aligned to the grid when it is turned on. Instead, 4th Dimension remembers each object's relation to the nearest intersection. If you then move or resize the object, 4th Dimension aligns the object to the nearest intersection of the grid using the original relation. To align these objects to the new grid, you use the **Align to Grid** menu item.

The figure below depicts the invisible grid used to align objects.



► To align existing objects to the grid:

1 **With the grid turned on, select the object or objects you want to align to the grid.**

2 **Choose the Align to Grid from the Object menu.**
OR
Click the Align to Grid icon .
4th Dimension aligns the upper-left corner of the object or of each object, to the nearest point on the invisible grid.

Distributing Objects

You can distribute objects so that they are aligned and have an equal amount of space between them.

You can distribute objects using either the Distribute tools in the Tools palette or the Alignment Assistant. The latter allows you to align and distribute objects in one operation.

► To distribute objects with equal spacing:

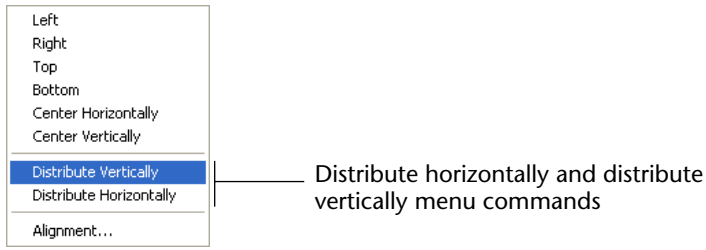
1 **Select three or more objects and click the desired Distribute tool.**
For more information on how to select objects, refer to "Selecting Objects" on page 302.

2 **Click the distribution tool that matches your needs.**

Distribute horizontally — Distribute vertically

OR
Select a distribution menu command from the Align submenu in the Object menu.
OR

Select a distribution menu command from the Align submenu in the contextual menu that is displayed when you right-click one object of the selection (Windows) or when you press the Control key while clicking an object of the selection.



4th Dimension distributes the objects accordingly. Objects are distributed using the distance to their centers and the largest distance between two consecutive objects is used as a reference.

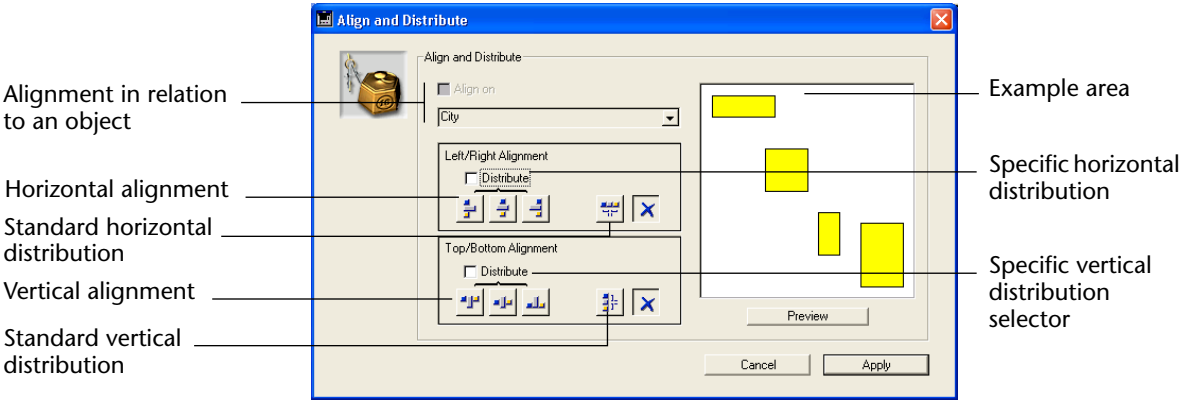
► To distribute objects using the Alignment dialog box:

1 **Select the objects you want to distribute.**

For more information on how to select objects, refer to "Selecting Multiple Objects" on page 304.

2 **Choose the Alignment command from the Align submenu in the Object menu.**
OR
Choose the Alignment command from the Align submenu in the Form editor contextual submenu.

The Alignment Assistant appears:



3 **In the Left/Right alignment and/or Top/Bottom alignment areas, click the standard distribution icon.**

The example area displays the results of your selection.

4 **To perform a distribution that uses the standard scheme, click Preview or Apply.**

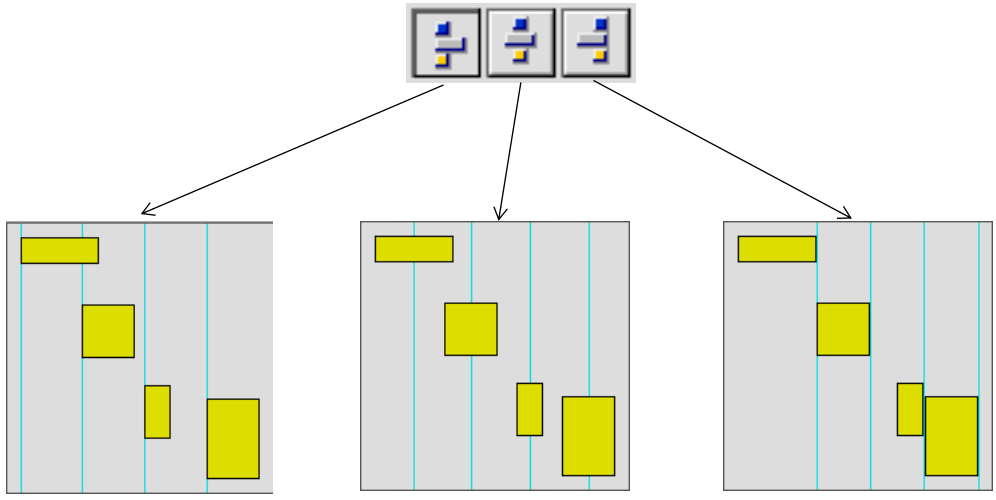
In this case 4th Dimension will perform a standard distribution, as described earlier in this section.

OR

To execute a specific distribution, select the Distribute option (for example if you want to distribute the objects based on the distance to their right side).

This option acts like a switch. If the Distribute check box is selected, the icons located above it perform a different function. They now allow you to specify whether the selected objects will be distributed evenly with respect to their left/top sides, their centers, or their right/bottom sides. The effects of each icon are shown in the preview area and are illustrated below.

- **Horizontal Distribution** For horizontal distribution (i.e., from left to right): the icons have the following meaning:

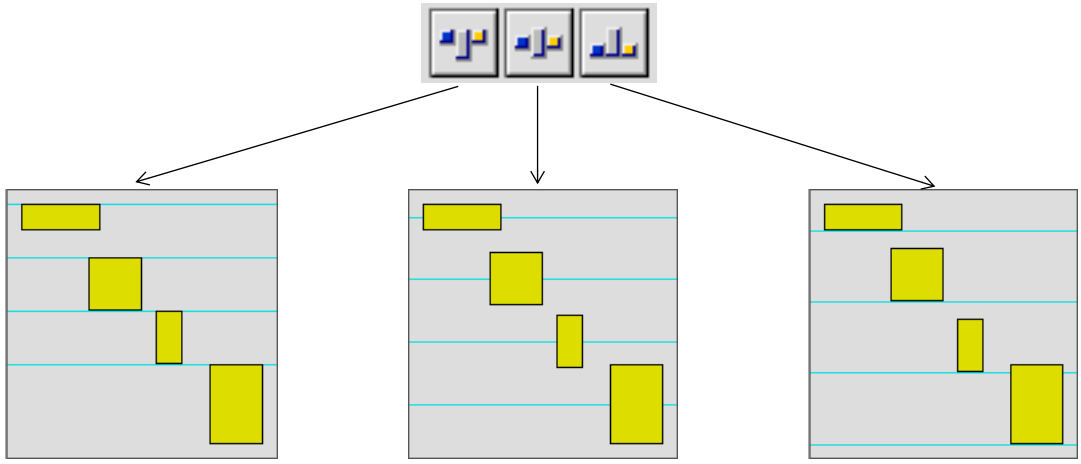


Distances between the objects' left sides are equal

Distances between the objects' horizontal centers are equal

Distances between the objects' right sides are equal

■ **Vertical Distribution** For vertical distribution (i.e, from top to bottom), the icons have the following meanings.

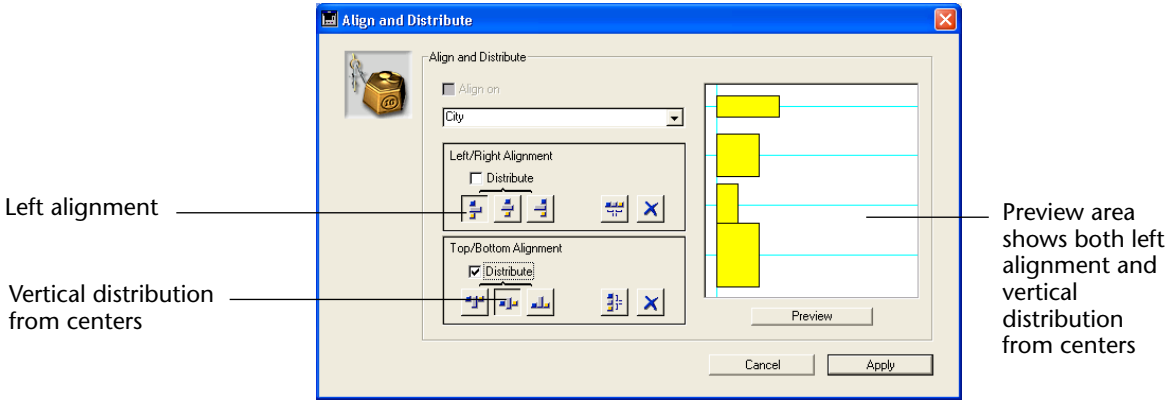


Distances between the objects' tops are equal

Distances between the objects' vertical centers are equal

Distances between the objects' bottoms are equal

The Alignment Assistant allows you to align and distribute objects in one operation. For example, suppose you have a group of field labels that are arranged vertically. You would like to left align the labels and distribute them evenly in the vertical dimension. The following specification does this:



Duplicating Objects

You can duplicate any object in the form, including active objects. Copies of active objects retain all the properties of the original, including name, type, standard action, display format, and object method.

You can duplicate an object directly using the Duplicate tool in the Tools palette or use the Duplicate Many dialog box to duplicate an object more than once. Also, using that dialog box, you can set the distance between two copies.

When duplicating a variable, you can use specific duplication features that allow you to include an automatic number in the copies' names. For more information on this point, refer to "Duplicating on a Matrix" on page 459.

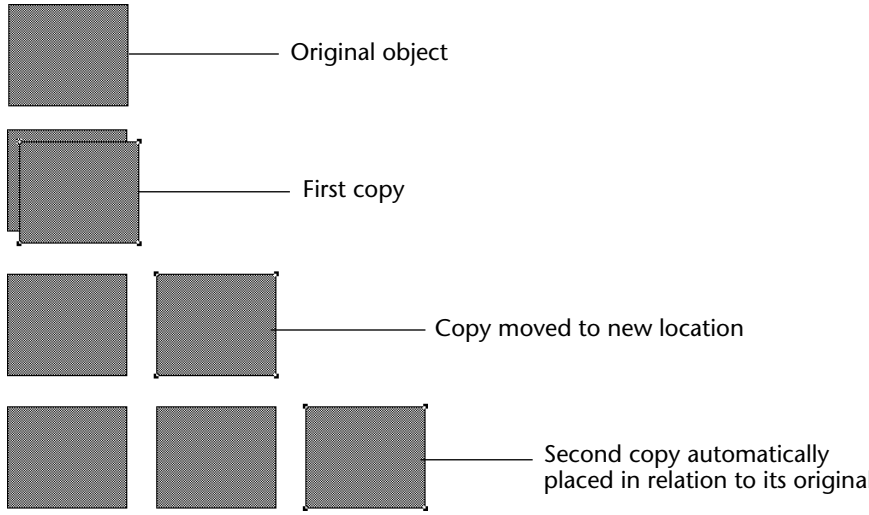
► To duplicate an object:

1 **Select the object or objects that you want to duplicate.**

2 **Choose Duplicate from the Object menu.**
OR
Click the Duplicate icon  **in the Tools palette.**
OR
Press the keys Ctrl+D (Windows) or Command+D (on MacOS).
4th Dimension creates a copy of each selected object and places the copy in front and slightly to the side of the original.

3 **Move the copy (or copies) to the desired location.**
If you choose the **Duplicate** menu item again, 4th Dimension creates another copy of each object and moves it the exact same distance and direction from the first copy.
If you need to distribute copies of the object along a line, you should use the following procedure. Duplicate the original object, move the copy to another location in the form, and then duplicate the copy. The second copy is automatically placed in the same relation to the first copy as the first copy was in relation to the original object. Subsequent copies are also placed in the same relation to their originals.

The figure below shows how this relative placement of copies works:

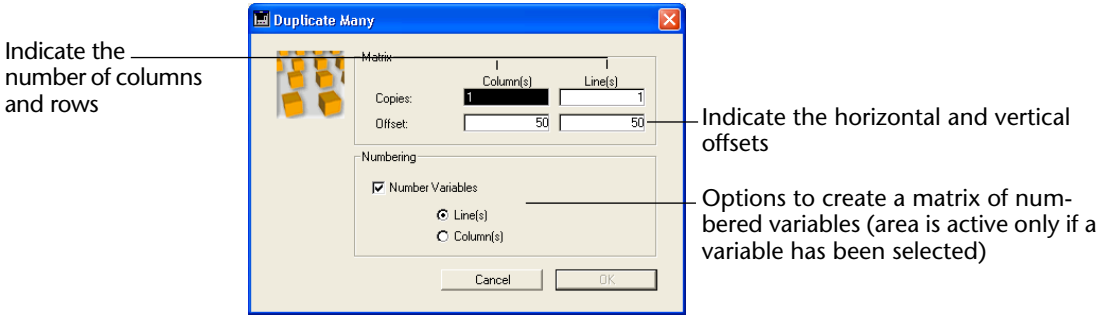


► To duplicate multiple objects using the Duplicate Many dialog box:

1 Select the object(s) to duplicate.

2 Choose Duplicate Many... from the Object menu.

The Duplicate Many dialog box appears:



3 In the upper area, enter the number of columns and lines (rows) of objects you want to get.

- For example, if you want three columns and two lines of objects, enter 3 in the Column(s) area and 2 in the Line(s) area.
- If you want three horizontal new copies of an object, enter 4 in the Column(s) area and leave the default value, 1, in the Line(s) area.

4 For lines and columns, define the offset that you wish to leave between each copy.

The value must be expressed in points. It will be applied to r each copy, or copies, in relation to the original object.
For example, if you want to leave a vertical offset of 20 points between each object and the height of the source object is 50 points, enter 70 in the column's "Offset" area.

5 **If you wish to create a matrix of variables, select the Number Variables option and select the direction in which the variables are to be numbered, either by line(s) or by column(s) (otherwise go to step 6).**

This option is active only if the selected object is a variable. For more information on that option, refer to "Duplicating on a Matrix" on page 459.

6 **Click the OK button.**

The number of columns and lines of the specified object(s) is created.

Copying Objects on a Form

You can copy all, some, or one of the objects on a form using the **Copy** menu item in the **Edit** menu. You can use the copied objects in another form in the database or in another database.

Like objects duplicated using the **Duplicate** command, each object copied using the **Copy** command is a complete copy of the original object and retains all the properties of the original. Copies of active objects such as fields and buttons retain all the properties of the original including name, type, action, display format, and the method associated with the object.

Objects copied using the **Copy** command are copied to the Clipboard while objects duplicated with the **Duplicate** command are reproduced on the current form page.

You can copy objects and save them in the Scrapbook for later use. For example, suppose you create several custom buttons. You can use the same buttons with their corresponding actions in any form in any database simply by copying them to the Scrapbook and then pasting them into another form.

Note A method that is copied with an object and pasted in a different form may lose its meaning unless it is updated. This is the case, for example, if the method refers to a specific field and the new form does not have this field.

► To copy all the objects in a form and paste them into a new form:

1 **Choose Select All from the Edit menu.**
4th Dimension selects every object on the current form page.

2 **Choose Copy from the Edit menu.**
4th Dimension places a copy of the form on the Clipboard.

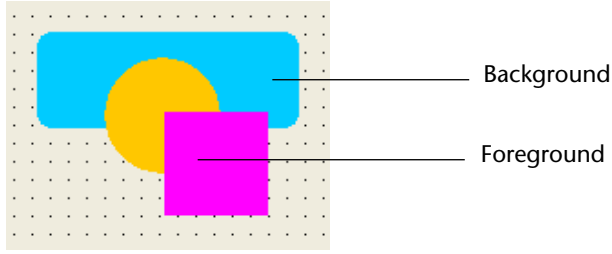
3 **Open a blank page or create a new form using the Form Wizard.**
For information on creating a form, see Chapter 4.

4 **Choose Paste from the Edit menu.**
The new form contains copies of all objects in the previous form.

Note If the form is to be used as an output form, you may need to adjust the output control lines. The output control lines are not associated with objects, so they must be repositioned on the new form. For information about working with output control lines, see "Moving Output Control Lines" on page 483.

Layering Objects

You will sometimes have to rearrange objects that are obstructing your view of other objects in the form. For example, you may have a graphic that you want to appear behind the fields in a form. 4th Dimension provides four menu items, **Move to Back**, **Move to Front**, **Up One Level** and **Down One Level** that let you "layer" objects on the form. The figure below shows objects in front of and behind other objects.



► To move one or several objects to the back:

1 **Select the object(s) that you want to move to the back.**

2 **Click the Move to Back icon  in the Tools palette.**
OR
Choose Move to Back from the Object menu.
OR
Press Ctrl+B (Windows) or Command–B (MacOS).
OR
Choose Move to Back from the Level submenu in the contextual menu. This menu is displayed by clicking the object using the right button of the mouse (Windows) or by clicking the object while pressing the Control key (MacOS).

4th Dimension moves the selected object or objects behind all the other form objects.

Note When you move an object to the back, it may be hidden by objects in front of it. To see the object, select the object in front and send it to the back.

► To move an object one level to the back:

1 **Select the object or objects that you want to move to the back.**

2 **Choose Down One Level from the Object menu.**
OR
Choose Move Down One Level from the Level submenu in the contextual menu. This menu is displayed by clicking the object using the right button of the mouse (Windows) or by clicking the object while pressing the Control key (MacOS).

4th Dimension moves the selected objects down one level.

► To move an object to the front:

1 **Select the object or objects that you want to move to the front.**

2 **Click the Move to Front tool  in the Tools palette.**
OR
Choose Move to Front from the Object menu.
OR
Press Ctrl+F (Windows) or Command–F (MacOS).
OR
Choose Move to Front from the Level submenu in the contextual menu. This menu is displayed by clicking the object using the right button of the mouse (Windows) or by clicking the object while pressing the Control key (MacOS).
4th Dimension moves the selected object or objects in front of all the other objects.

► To move an object one level to the front:

1 **Select the object or objects that you want to move to the front.**

2 **Choose Up One Level from the Object menu.**
OR
Choose Move Up One Level from the Level submenu in the contextual menu. This menu is displayed by clicking the object using the right button of the mouse (Windows) or by clicking the object while pressing the Control key (MacOS).

4th Dimension moves the selected objects up one level.

Deleting Objects

You can delete any object on a form. If you want, you can place a copy of the deleted object on the Clipboard. Objects placed on the Clipboard can later be pasted to new locations in the form. If the Object Properties window is open while you try to delete an object, the deletion will affect the contents of the Object Properties window.

► To delete an object:

1 **Select the object or objects you want to delete.**

2 **Choose Clear from the Edit menu.**
OR
Press the Backspace key (Windows) or the Delete key (MacOS) on your keyboard.
4th Dimension deletes the selected object or objects.

To cut an object to the Clipboard, choose **Cut** from the **Edit** menu. 4th Dimension removes the selected object or objects and places a copy on the Clipboard. The Cut command works even if the Object Properties window is open.

If you change your mind, choose **Undo** from the **Edit** menu or from the editor's contextual menu before performing another action. 4th Dimension restores the deleted object or objects.

Optimizing the Appearance of Text and Picture Objects

You can resize static text areas and pictures for optimal display based on their current graphic characteristics (font size, style, etc.).

To do this:

1 **Right-click (under Windows) or Control+click (under MacOS) on the object and choose the Automatic Size command from the contextual menu.**
OR
Hold down the Ctrl key (under Windows) or Command key (under MacOS) and click on the lower right corner of the object.

- Static text areas, check boxes, radio buttons and buttons will be resized so that their contents correspond exactly to their boundaries.
- Fields and variables will be resized so that their height is at least equivalent to that of the current font and their width is 100 points — if the size of the object is greater than these minimum sizes, the command will have no effect.
- Pictures or picture buttons will be displayed using the original size of the source picture.
- Scrollable areas and hierarchical lists will be displayed with a height in keeping with that of the current font.
- Combo boxes, pop-up/drop-down lists and hierarchical pop-up menus will be displayed with their default minimum height if the current height is insufficient.

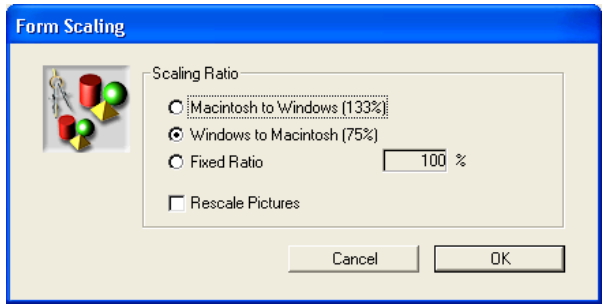
Scaling a Form

The Form editor includes a feature for rescaling form objects so that they look good when a database is transported to another platform.

Form objects created under MacOS will look smaller when viewed under Windows, and vice versa — even though the objects are actually the same size. This is because the Windows screen resolution is about 25% greater than the Macintosh resolution. For instance, 12-point text on a Macintosh will appear as 9-point text under Windows. If the font size is just large enough under MacOS, it may be too small under Windows. Conversely, if a font size under Windows is adequate, it may be too large under MacOS.

To compensate for screen resolution differences, you need to rescale objects. With the **Scale** item on the **Form** menu you can proportionally resize all the form objects in one operation.

When you choose **Scale**, the Form Scaling dialog box appears.



You can choose among the following options:

- **Macintosh to Windows platform (133%)** This option is the default option when you use 4th Dimension under MacOS. Use this option when you want to resize a form that was created according to the Macintosh screen resolution so it will look like it was created according to the Windows screen resolution. To do so, the program increases the size of all the form objects by one-third. For instance, 9-point text will become 12-point text.
- **Windows to Macintosh platform (75%)** This option is the default option when you use 4th Dimension under Windows. Use this option when you want to resize a form that was created according to the Windows screen resolution so it will look like it was created according to the Macintosh screen resolution. To do so, the program decreases the size of all the form objects by one-quarter. For instance, 12-point text will become 9-point text.

- **Fixed Ratio scaling** This option lets you resize a form using the percentage you type in the "%" enterable area. With this option you can resize a form so it will look good on any unusual screen resolution you may encounter on either the Macintosh or Windows platform. You can also use this option to change the size of all the form's objects for the platform you are using. For example, it you want to double the size of all objects, enter 200%; if you want to halve the size, enter 50%.
- **Rescale pictures** This option is not selected by default. Usually, decreasing or increasing the size of bitmapped pictures does not provide good results from a cosmetic point of view. For this reason, the program, does not resize any static pictures in a form unless you select this option. Instead it moves them to their new "center relative" positions. If you know that rescaling bitmaps will produce pleasing results or if you use non-bitmapped pictures, you may chose to rescale the pictures.

  When you have selected your options, click **OK** to resize the form, otherwise click **Cancel**. If you resize a form by mistake or with the wrong option, choose **Undo** from the **Edit** menu to recover the form as it was before the rescaling.

Changing the Appearance of Objects

You can change the appearance of any object in a form. Each object has its own platform interface and appearance setting.

For any object that uses text (a field, a text area, a button, and so forth), you can change the following attributes:

- Platform interface
- Appearance
- Font
- Style
- Size
- Alignment within the object's area.

For any object that uses lines, fill patterns, or colors, you can change the following attributes:

- Line width
- Fill pattern

- Border pattern
- Foreground and background color.

Platform Interface and Border Line Style

You can set the platform interface and appearance on an object-by-object basis. For platform interface, you have the following choices:

- **Inherited from Form** The platform interface for the object is the same as the platform interface of the form. The platform interface of the form is set in the Form Properties window.
- **Automatic** The platform interface is based on the one set using the **Auto Platform** menus (Mac and Windows) in the Preferences of the application (refer to the paragraph "Look Page," page 110).
- **MacOS 7** The object will be displayed as a Macintosh System 7 object.
- **Windows 3.11, NT 3.51** The object will be displayed as a Windows 3.11 or NT 3.51 object.
- **Windows 95/98/2000, NT 4** The object will be displayed as a Windows 95/98/2000 or NT 4 object.
- **Mac OS 9** The object will be displayed as a Macintosh object respecting the Mac OS 9 graphic interface guidelines.
- **Mac Theme**: The object will be displayed using the theme defined under the MacOS "Appearances" control panel.

  For information on setting the Platform Interface for the form, see "Setting the Platform" on page 292. For information on each platform interface option, see the paragraph "Platform Interface" on page 98.

  For Border Line Style, you have the following choices:

  - Transparent
  - Plain
  - Dotted
  - Raised
  - Sunken
  - Double.

  For illustrations of the effects of these choices on various object types, see the sections "Fields and Field Labels" on page 258 and "Buttons" on page 421.

Platform interface and appearance are set using the Display page of the Object Properties window.

► To set Platform Interface or Appearance:

1 **Double-click the object.**
Depending on your display options, the Object Properties window or the Property List for the object appears.

2 **In the Object Properties window, click the Display tab.**
OR
In the Property List, expand the Appearance theme.

3 **Choose the desired platform interface and border line style from the drop-down lists.**

Working with Text Areas

You can make the following changes to text areas:

- Establish default settings for font attributes,
- Create text areas and add or edit text,
- Setting the text area's platform interface, appearance, and font attributes.

Creating and Editing Text Areas

Use text areas to provide labels, titles, and descriptions in your form.

Text you create in a text area is different from the text contained in a Text field. A Text field contains data stored in the database. The contents can be different in each record. A text area is a graphic object; it is not active. Text in a text area remains the same whenever the form is displayed.

There are three exceptions to this rule:

- You can embed field names or variable names in text areas. When the text area is displayed or printed, the values of the fields or variables from the current record are substituted. Use embedded fields and variables to create mail-merge documents and in report headers and footers. For complete details, see the section "Creating Mail-Merge Documents" on page 493.

- You can integrate dynamic table or field references to text areas. When you place dynamic labels in your forms, they are automatically updated throughout the database when you modify a table's or a field's name using the Structure editor or the Table name and Field name commands.

  To insert a dynamic table name: <**?[TableName]**> or <**?[2]**> (the table's creation order number, meaning the second table created).

  To insert a dynamic field name: <**?[TableName]FieldName**> or <**?[2]3**> (the table's and field's creation order number), or even <**?3**> (the field's creation order number) for the current table's field.

  Please note that table and field numbers correspond to their creation order. You can therefore add or rename tables and fields without modifying the dynamic reference system. The actual contents of the text area can be displayed using the **Show Name** and **Show Format** menu commands from the Object menu.

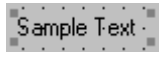
Note You can automatically insert dynamic references using the Options page of the Form Wizard. For more information about this point, refer to "Form Display Options" on page 254.

- You can assign a STR# resource to a Text area. Use the format *:STR# ID, line ID* in place of static text. If, for example, you enter *:20000,10*, the reference will be replaced by the text whose ID is 10 from the resource whose ID is 20000. When you choose **Show Resource** from the **Object** menu, the text in the resource appears in the Form editor.

  The Form Wizard automatically creates text areas that contain field labels for the fields and, optionally, a title for the form. You can modify or create these labels as you would modify or create any other text area you add to the form.

  As you draw a text area, it snaps to a size that reflects the font size.

  After you create a text area, the text box displays a sample text that is selected.



Type text in the text area. When the text you are typing reaches the edge of the text area, 4th Dimension automatically wraps the text to the next line in the area.

Note You can tell 4th Dimension to resize a text area to fit its contents. **Ctrl+click** (Windows) or **Command-click** (MacOS) the bottom right corner of the object to resize the object.

If you enter more text than the area can display, the text is not visible until you resize the text area.

► To modify text in a text area:

1 **Click the Text Area tool**  .

2 **Select the text you want to modify or click to insert an insertion point.**
Use the standard text editing operations to edit the text.

Default Settings for Text Objects

When you create any object, 4th Dimension uses the default settings for these features. You can establish new default settings at any time.

For example, if you establish a new default font, 4th Dimension uses that font for any object you subsequently create that displays text.

You can change the settings for a selected object without changing the default settings. 4th Dimension changes the appearance of the selected object, but the change affects only that object. 4th Dimension continues to use the default settings for objects created in the form.

For example, if you change the font for one text area, the change affects only that area, not subsequently created fields or text areas.

This section gives the basic steps for establishing default settings and for changing the settings for selected objects. It then describes in detail each option for the appearance of objects in a form.

► To establish default settings:

1 **Make sure that no object in the form is selected.**

2 **Use the items in the Font, Style, and Object menus to choose the font, size, style, text justification, line width, fill pattern, border pattern, or color.**
These settings will be used as the default settings for any new objects that use text that you create.

The following sections describe how to change these attributes for selected objects.

Setting a Text Object's Platform Interface and Appearance

You can set the platform interface, appearance, color, border, and fill pattern of a text object.

► To set the text object's appearance:

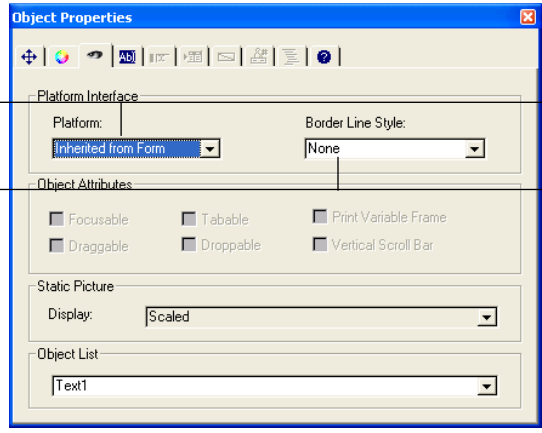
1 **Double-click the text object to display its properties in either the Object Properties window or the Property List.**

The choice between these two palettes is explained in "Displaying and Setting Form and Object Properties" on page 282.

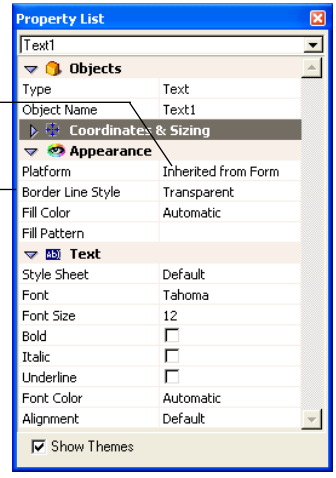
2 **In the Property List, expand the Appearance theme.**
OR
In the Object Properties window, click the Display tab control.

The appearance properties for the object are displayed:

Object Properties Window



Property List



Platform interface

Border style

3 **Use the drop-down lists to set the desired appearance of the object.**

For information on these options, see the section "Platform Interface and Border Line Style" on page 335.

Note You can also set the border line style using the Form editor contextual menu.

Setting Text Attributes

You can set text attributes for text objects using either the Object Properties window or the Font and Style menus.

► To set text attributes using the Font and Style menus:

1 **Select the object or objects whose text attributes you want to change.**

2 **Make the appropriate selections from the Font and Style menus.**

► To set text attributes using the Object Properties window or the Property List:

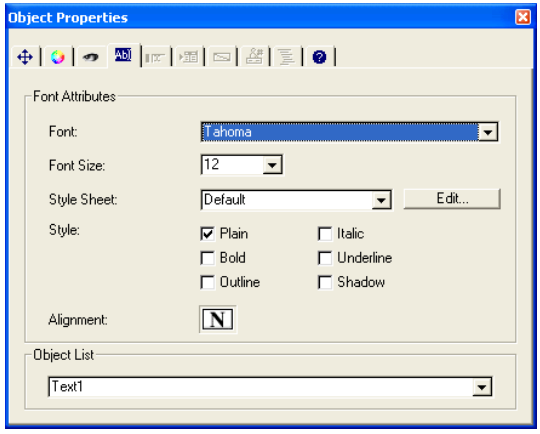
1 **Double-click the picture to display its properties in either the Object Properties window or the Property List.**

The choice between these two windows is explained in "Displaying and Setting Form and Object Properties" on page 282.

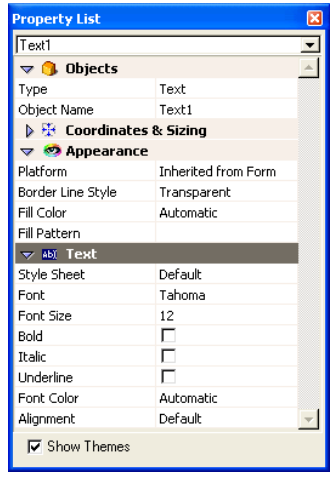
2 **In the Object Properties window, Click the Font tab to view the current text attributes.**
OR
In the Property List, expand the Text theme.

The text properties are displayed:

Object Properties Window



Property List



3 **Make the appropriate font, font size, and font style selections.**
OR
Choose a style sheet from the Style Sheet drop-down list.

4 **Choose an alignment.**

5 **To set text attributes for another text object, choose the desired object from the Object List or click another object on the form.**

The settings in the Font page change to reflect the settings of the new object. If you select several text objects, the Object List area changes to "Selected Objects."

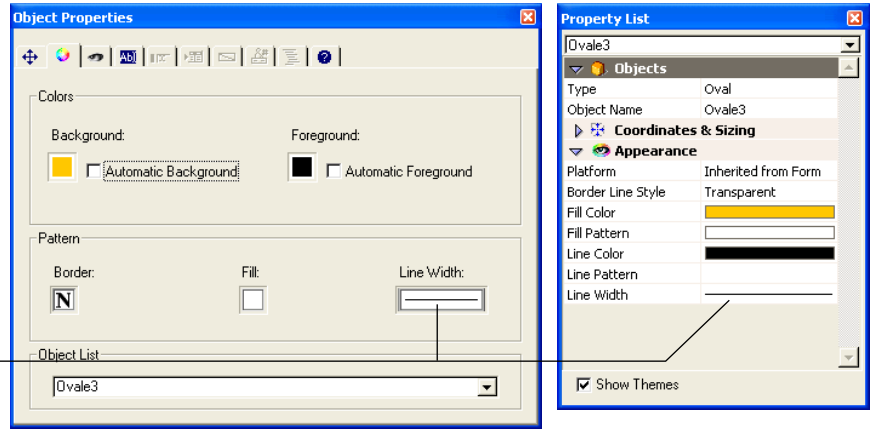
Line Widths

4th Dimension lets you specify different widths for lines and objects that have lines such as ovals, grids, and rectangles.

You can specify line widths using either the **Line Width** menu item on the **Objects** menu, the Colors page of the Object Properties window or the Appearance theme of the Property List.

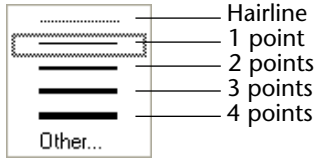
Object Properties Window

Property List



Line width menu

Choose one of the line widths. The first choice is the hairline, i.e., the thinnest line that can be printed by the printer, followed by 1, 2, 3 or 4 point lines.



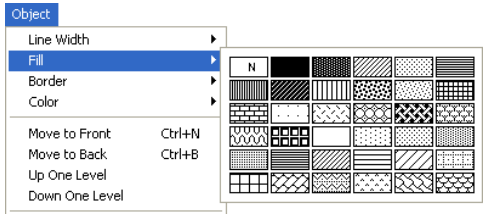
If you choose **Other**, 4th Dimension displays a dialog box in which you can specify any line width up to 20 points.

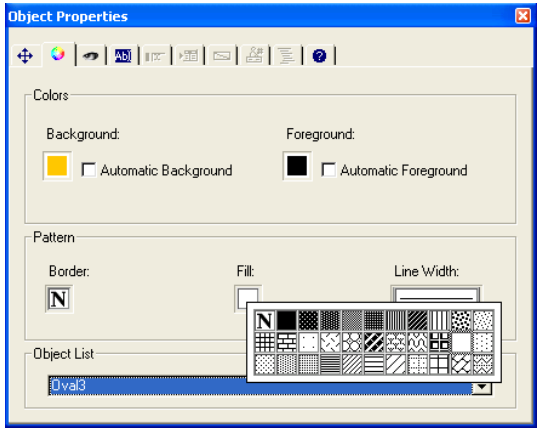
Fill Patterns

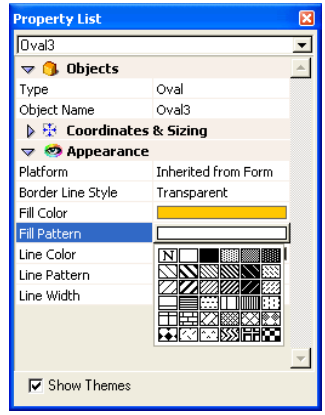
You can apply a fill pattern to any two-dimensional graphic object in the form such as an oval, a rectangle, a line, a grid object, and the enclosed area of a text object.

You can choose the fill pattern using either the **Fill** menu item in the **Object** menu or the Fill Pattern pop-up menu in the Colors page of the Object Properties window.

Select an object and choose one of the patterns displayed in either **Fill** menu item or from the Property List.





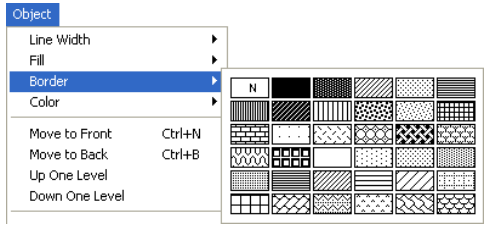


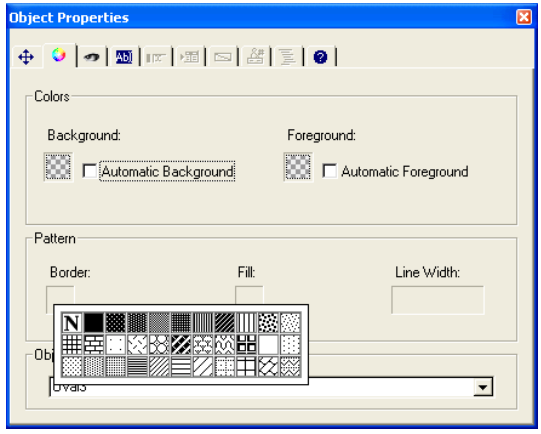
Border Patterns

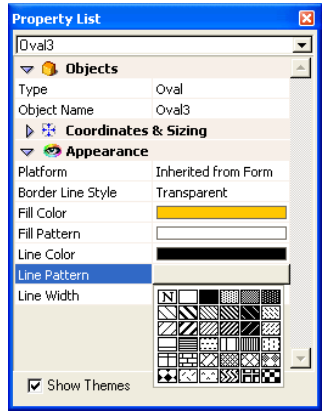
You can set patterns for the borders of any object in the form that has a border — such as an oval, a rectangle, and a grid object. The border patterns available are the same as the fill patterns. The appearance of the border also depends on the line width you have specified for the border.

You can set the border pattern using either the **Border** item of the **Object** menu, the Border menu in the Colors page of the Object Properties window or the Property List.

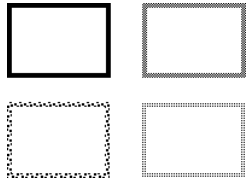
Choose one of the patterns displayed in the **Border** menu item.







The following illustration shows examples of borders.



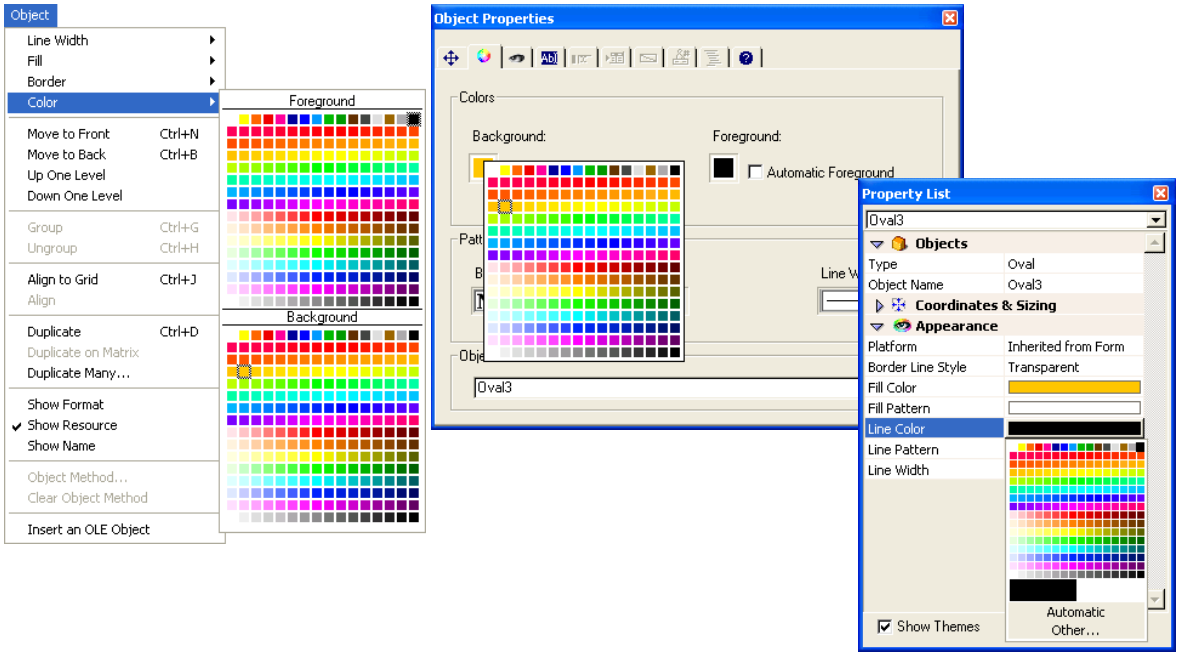
Foreground and Background Colors

4th Dimension lets you add colors to objects for display on a color monitor or (if your printer supports color) for color printing. By combining colors and patterns your possibilities are virtually endless.

You can specify different colors for foreground pixels and background pixels. On a color display, the mixing of foreground and background colors allows you to create custom tints and shades.

In the Property List, the background color is called Fill Color and the foreground color is called Line Color.

You can set foreground and background colors using either the Colors page of the Object Properties window, the **Color** item in the **Object** or contextual menu, or the Property List.



Note • If you select the **Automatic** option for foreground and background colors, the colors used will be the colors selected in your OS.

You can select the **Automatic** option by selecting Foreground or Background in the **Color** submenu of the **Object** or contextual menu.

• The Property List allows you to use the system color chooser to define a color by selecting **Other**.

If your monitor supports only 16 colors, choose colors from the first 16 colors on each palette. If your monitor supports 256 colors (or more), any colors you choose will display properly.

Placing a Picture from the Picture Library

You can insert static pictures in your forms using two methods:

- By pasting a picture,
- By dragging a picture from the Picture Library.

The Picture Library stores images that you can use as graphic elements on forms, as picture menu items, as picture buttons, as small icons in lists, and as custom toolbar icons. When you want to place a background graphic on a form, you should add it to the Picture Library and then place the picture on the form. If you use a picture in the Picture Library on more than one form, it is stored only once. Also, if you update a picture in the Picture Library, references to it will be updated automatically throughout the database.

For more information on how to use the Picture Library refer to Chapter 12.

If you place a picture on page 0 of a multi-page form, it will appear automatically as a background element on all pages. You can also include it an inherited form, applied in the background of different other forms. Therefore, your database will run faster than if the picture was pasted into each page.

► To place a picture on a form:

1 **Open the form in which you want to place the picture.**
If necessary, navigate to the page on which you want to place the picture. For information on navigation, see the section "Moving from Page to Page" on page 352.

2 **Open the Picture Library and click on the name of the desired picture.**
OR
If the picture you want to insert is on the Clipboard, choose Paste from the Edit menu and then go to step 6.

3 **Display the pictures as a list in the Picture Library.**

4 **Click on the picture and drag it from the Picture Library to the form.**

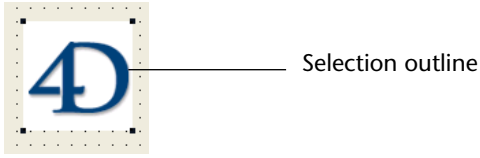
- If the picture you drag is defined as a table of thumbnails, it will automatically be inserted as a picture button (or picture pop-up menu if you hold down the **Shift** key when dragging the picture). If you want to insert it as a static or library picture, press the **Alt** key (Windows) or the **Option** key (MacOS) when dragging the picture.
- By default, the inserted picture will be of the **Library Picture** type, i.e., 4th Dimension will maintain the link with the original picture of the library. If you want to break this link, you must change the inserted picture to the **Static Picture** type using the Property List (see the paragraph "Dissociating a form Picture from its Library Source," page 349).
  The properties of static pictures and library pictures are generally identical.

5 **When it is at the desired location, release the mouse button.**

6 **Reposition the picture as desired and set its properties.**
The picture has a set of object properties, just as any object on the form. If you like, you can modify those properties. This is described in the following sections.

Modifying the Background of the Picture

You can change the picture's background to Transparent so that it takes the form's background color.



▶ To modify the picture's background:

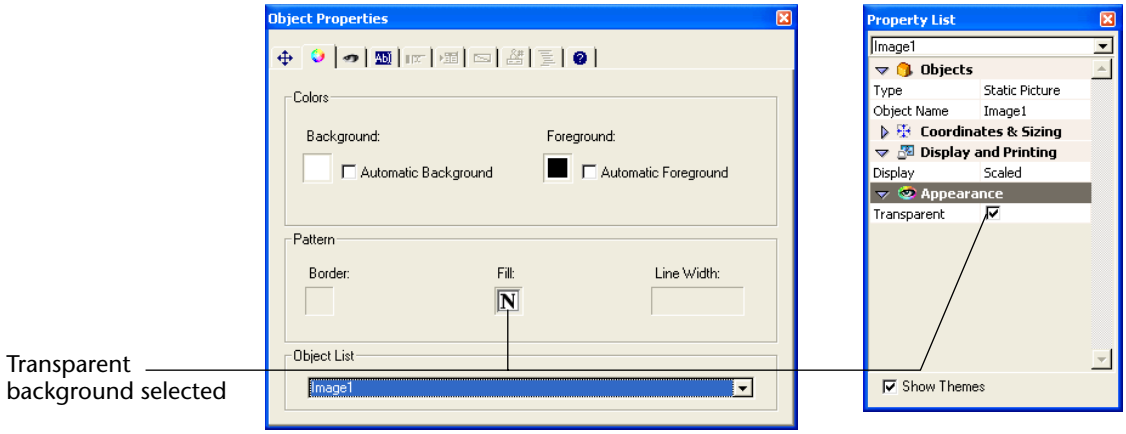
1 **Double-click the picture to display its properties in either the Object Properties window or the Property List.**
The choice between these two palettes is explained in "Displaying and Setting Form and Object Properties" on page 282.

2 **In the Property List, expand the Appearance theme.**
OR
In the Object Properties window, click the Colors tab.

3 **In the Object Properties window, select the N character in the Fill picture menu.**
OR
In the Property List, check the "Transparent" option.

Object Properties Window

Property List



The background of the picture becomes transparent:

Setting the Display Mode for a Static Picture

You can set the display mode of a static or library picture that is placed in a form.

► To set the display mode for a static or library picture:

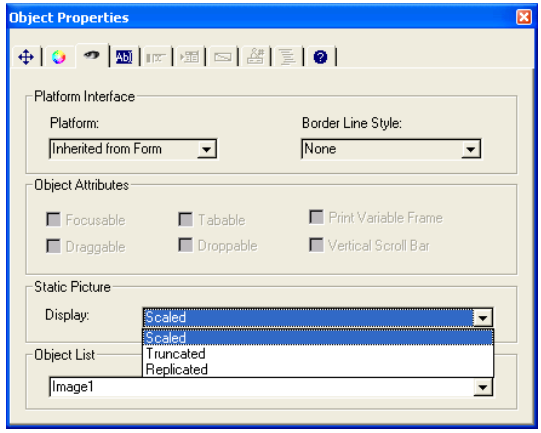
1 **Double-click the picture to display its properties in either the Object Properties window or the Property List.**

The choice between these two palettes is explained in "Displaying and Setting Form and Object Properties" on page 282.

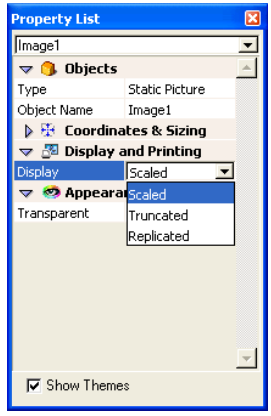
2 **In the Property List, expand the Display and Printing theme.**
OR
In the Object Properties window, click the Display tab.

3 **Choose a display option from the Display drop-down list.**

Object Properties Window



Property List



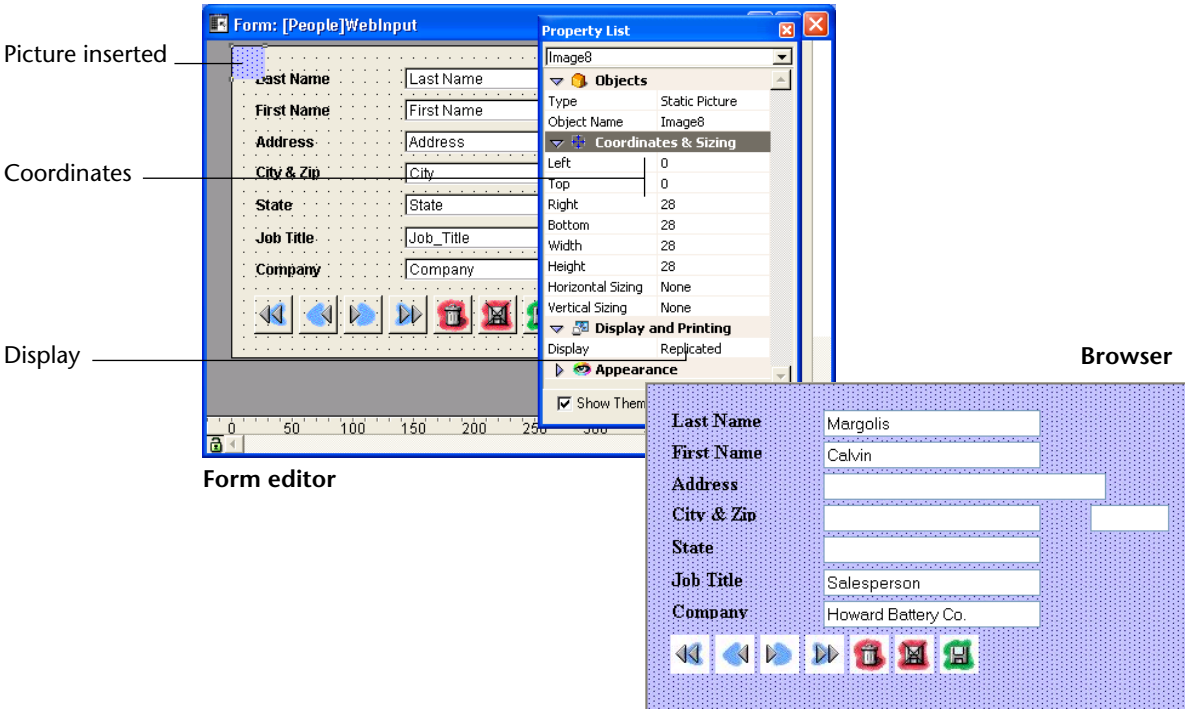
- **Scaled (default mode)** When the picture object is resized, the picture is resized so that the entire picture remains visible.
- **Truncated** When the picture object is resized, the picture keeps its proportions and only its boundaries change. The picture always stays in the centre of the picture object. If the picture object is reduced to a smaller size than the picture, the picture is truncated.
- **Replicated** When the size of the picture object is increased, the picture is replicated as many times as necessary to fill the new area. Recommended for background pictures since it doesn't require a lot of memory. The Form wizard uses this option when selecting the **Background picture** option (please refer to "Form Display Options" on page 254). If the picture size is reduced to a smaller size than the original picture size, the picture is truncated (not centered).

Defining a Background Picture for Web forms

You can insert a static picture in a form designed to be published on the Web and use that picture as a background replicated picture for browsers. To do so, the picture settings must meet the following conditions:

- The picture must be located in the upper left corner of the form (coordinates (0,0,x,x).

- Replicated option has to be assigned to the Display property of the picture.



Dissociating a form Picture from its Library Source

When you insert a picture that comes from the Picture Library, you actually insert a reference to a picture. The Property List will indicate its Type (under Objects) as a Library Picture. If the picture is modified in the Picture Library, each instance of it will be modified accordingly.

You may want to dissociate a picture inserted in a form from its source in the Picture Library.

► To disassociate a picture from its source in the Picture Library.

1 **Display the picture's properties in the Property List.**

2 **If necessary, expand the Objects theme.**

3 **Click the Type line.**
The Property List displays two types: Library Picture and Static Picture.

4 **Select Static picture.**
The picture is then treated as if it had been pasted from the Clipboard.

Creating a Multi-page Form

You can create multiple pages for an input form. If you have more fields than will fit on one screen, you may want to create additional pages to display them. Multiple pages allow you to do the following:

- Place the most important information on the first page and less important information on other pages,
- Organize each topic on its own page,
- Reduce or eliminate scrolling during data entry,
- Provide space around the form elements for an attractive screen design.

Note Multiple pages are a convenience used for input forms only. They are not for printed output. When a multi-page form is printed, only the first page is printed.

There are no restrictions on the number of pages a form can have. The same field can appear any number of times in a form and on as many pages as you want. However, the more pages you have in a form, the longer it will take to display the form.

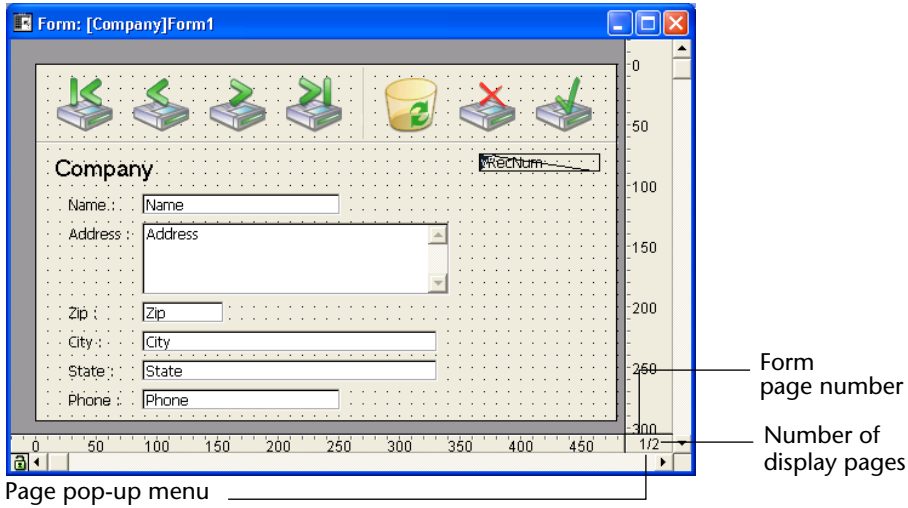
A multi-page form has a both a background page and several display pages. In the Form editor, objects that are placed on the background page may be visible on all display pages, but can be selected and edited[1] only on the background page. In multi-page forms, you should put your button palette on the background page. You also need to include one or more objects on the background page that provide page navigation tools for the user. For information on adding page navigation tools, see the section "Adding Page Navigation Controls" on page 354. This section details how to add and delete pages, add objects to the background page, move from page to page, and add fields to a new page.

Adding a Display Page to a Form

Every form has at least one display page[2] and a background page. The current page number appears in a box in the lower-right corner of the form window when rulers are displayed. This corner also includes a pop-up menu that allows you to select the displayed page.

1. In the User and Custom Menus environment, objects placed on the background page are always visible on all the pages.

The background page is numbered zero (0).



► To add a display page:

1 **Move to the last page of the form, then click the Next Page icon in the Tools palette.**

4th Dimension displays a dialog box asking if you want to add a page. Click **OK** to create the new page.

OR

Choose Add Page from the Form menu.

4th Dimension creates a new page.

A new, blank display page appears in the Form editor window. The page indicator box in the lower-right corner of the window displays the number of the page you are viewing.

You can now add fields and other form elements to the new page.

► To insert a display page before the current page:

1 **Choose Insert Page from the Form menu.**

2. The Options page of the Form Wizard contains an option that instructs the Form Wizard to create a multi-page form automatically if the fields you select don't fit on one page. If you selected this option, your form may initially have more than one display page. When the Form Wizard creates more than one display page, it puts buttons, variables, the form title, and decorative rectangles on the background page.

4th Dimension inserts a new page before the current page and displays it on screen.

Moving from Page to Page

When you want to display the background page or move to another display page, you can either use the page navigation tools in the Tools palette or the page pop-up menu in the Form editor window.

► To display the background page (page 0):

■ **Move to the first page of the form and click the Previous Page icon  in the Tools palette.**
OR
Use the Page pop-up menu to move to page 0.
OR
Select 0 from the Goto Page submenu in the Form menu or in the Form editor contextual menu
OR
Press Alt+click (Windows) or Option+click (MacOS) on an object belonging to page 0, or specifically outside any object belonging to the current page[1].
4th Dimension displays the background page. The page number of the background page is zero (0).Objects located in the background page are displayed in each page. You can place any object type in the background page.

Note There should be no confusion between using the **Page 0** menu item from the **Display** submenu and actually making the background page the current edited picture. Selecting the **Page 0** menu item from the **Display** submenu only displays the items of the background page. For more information, refer to "Showing/Hiding Elements in the Form Editor" on page 270.

If you want to use a graphic as a background image, add it to the Picture Library and then place it on the background page. It is also possible to paste the picture directly into the form. For information on the using the Picture Library, refer to the chapter "Using the Picture Library," page 659.

1. The **Alt+click** or **Option+click** shortcuts on an object on the current page creates or opens a method object.

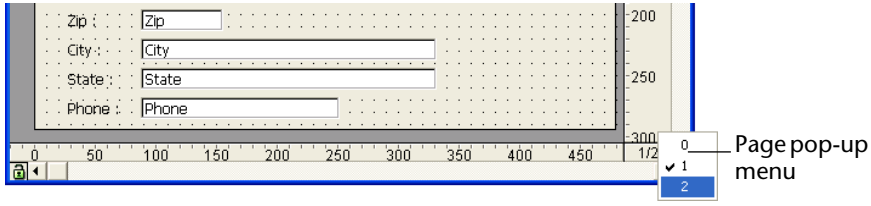
► To use the page navigation tools:

- **To move to the next page, click the Next Page icon in the Tools palette.**
- **To move to the previous page, click the Previous Page icon in the Tools palette.**
  4th Dimension displays the page immediately following or prior to the current page.

  If you click **Previous Page** while viewing the first page of the form, the background page appears. If you click **Previous Page** while viewing the background page, nothing happens. If you click **Next Page** while viewing the last page of the form, 4th Dimension asks if you want to create another page for the form.

► To display any page:

1 **Hold down the mouse button on the page indicator at the bottom-right corner of a Form editor window.**



OR
Display the Goto Page submenu in the Form editor contextual menu.

2 **Choose the desired page number.**

Deleting a Page

You can delete unwanted display pages from a multi-page form. Any fields or other objects on the deleted pages will be deleted as well. The remaining pages are renumbered. You cannot delete the first page or the background page in a form that consists only of those two pages.

► To delete a page from the form:

1 **Use either the page navigation tools or the page pop-up menu to display the page you want to delete.**

2 **Click the Delete Page tool in the Tools palette.**
OR
Choose Delete Page from the Form menu.

A dialog box appears verifying that you want to delete the page from the form.

3 **Click OK.**

The page and any objects on the page are removed from the form.

Adding Fields to a Blank Page

When you add a new page to a form, it is blank. You can add fields in the following ways:

- Use the Add Field tool to place each field.
- Copy or duplicate fields from other pages or from other forms, paste them onto the new page, and change the copied fields' properties.
- Drag fields from the Tables page of the Explorer onto the form.

Adding Page Navigation Controls

When you create a multi-page form, you need to provide a way for users to move from one page to another. 4th Dimension provides three ways that you can use to add navigation tools:

- **Tab control** The tab control object gives users random access to individual pages. You place the tab control on the background page of the form and use its properties to provide page navigation controls.
- **Automatic buttons** You can add automatic page navigation buttons to the form — First Page, Last Page, Previous Page, and Next Page. These buttons should be placed on the background page.
- **Object methods** In addition, the language includes the GOTO PAGE command. You can use this command as part of an object method to create custom navigation controls using any suitable object type. For example, you can choose to use a picture button or pop-up menu to provide page navigation controls.

Adding Page Navigation Buttons

You can include page navigation buttons when you generate the form using the Buttons page of the Form Wizard. After the form is generated, open it in the Form editor and add the necessary pages. If you need to add the page navigation buttons after the form is created, you can do so using either the Active object tool in the Tools palette or a button tool in the Objects palette. For more information, see "Creating an Active Object" on page 414.

Using a Tab Control

The tab control provides a visual indication of the current page and the remaining pages. For information on creating and activating a tab control, see the section "Tab Controls" on page 445.

Inherited Forms

4th Dimension allows you to use "inherited forms." The principle behind this new feature is to be able to use objects from Form A in a Form B: Form B "inherits" the objects from Form A.

Suppose, for example, that all entry forms belonging to a database have to contain the **OK**, **Cancel**, **Next**, **Previous** buttons as well as a logo. Simply create a form containing only these elements and then call it as an inherited form in all database entry forms. Each entry form contains only fields and objects specific to its use.

Unlike form "models" defined using the form wizard (see the paragraph "Creating a Form Template", page 258), the reference to the inherited form is always active: if an element of the inherited form is modified (button styles, for example), all forms using this element will automatically be modified.

Using inherited forms

Once using the database, inherited form objects are dynamically combined with those of the open form. This mechanism is very similar to that of the "page 0" form mechanisms, the difference being that it can be applied to the database form set.

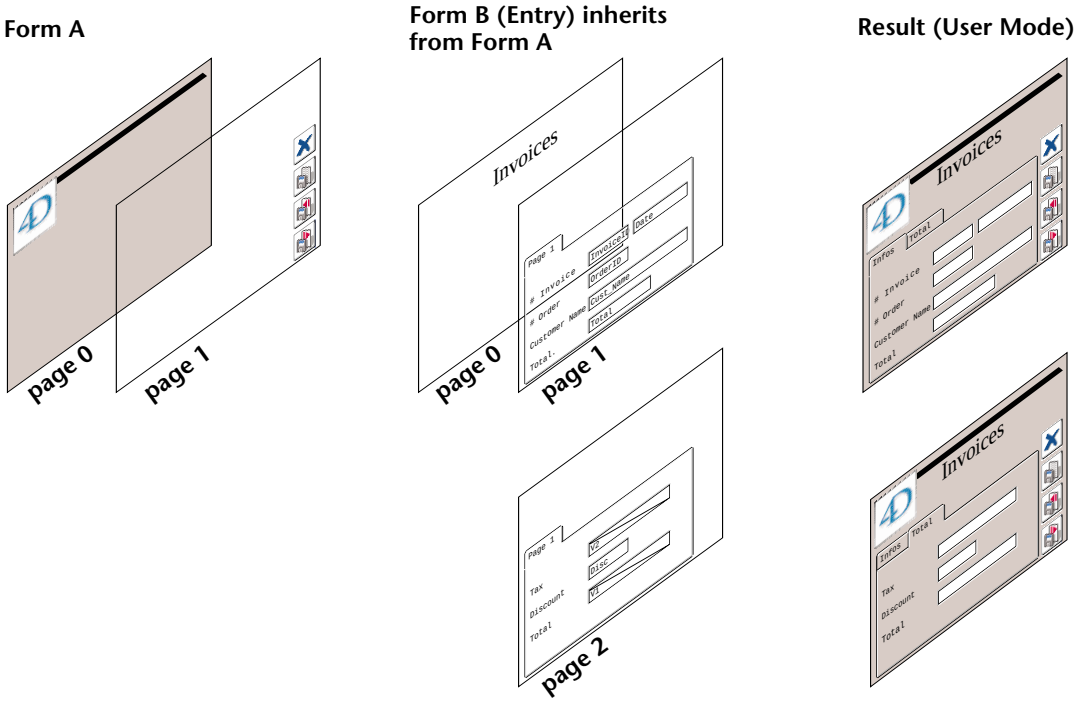
Once you open a form in the User environment or Custom menus, the objects are loaded and combined in the following order:
1- Page zero of the inherited form
2- Page 1 of the inherited form
3- Page zero of the open form
4- Current page of the open form

This order determine the entry order of objects in the form.

Note Only pages 0 and 1 of the inherited form can appear in other forms.

The properties (window name, re-sizing, events, etc.) and the method of an inherited form are not considered when used as an inherited form. On the other hand, the methods of objects that it contains are called.

The following diagram illustrates how inherited forms work



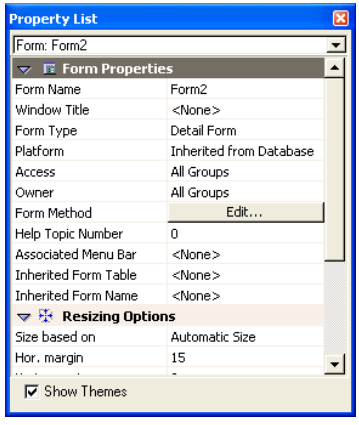
Defining an inherited form

Form inheritance starts in the 4D Form editor.

► To define an inherited form:

1 **In the Form editor, open the form before inheriting another form.**

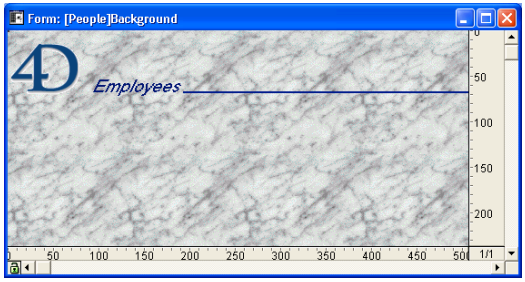
2 **Display the Property List and click outside all objects in the form in order to see form properties.**



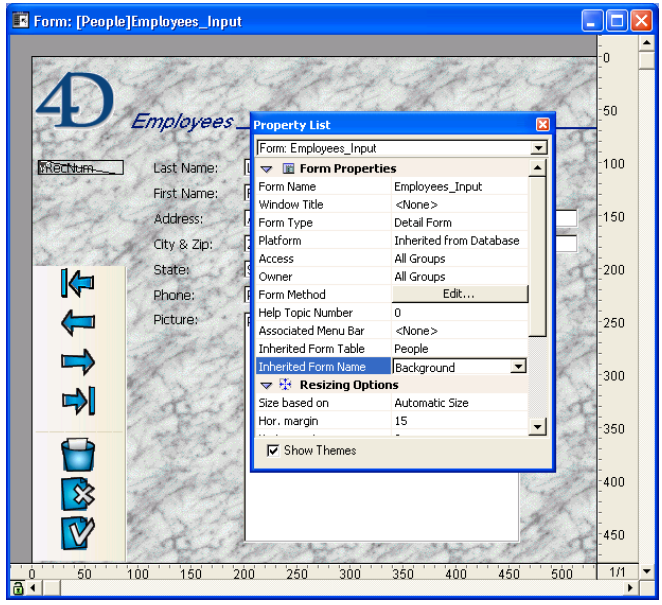
The “Inherited form table” and “Inherited form name” are available. All database tables and their forms are displayed.

3 Select the table and then the form name to inherit.

All forms can be designated as inherited form. However, the contained elements must be compatible with use in different database tables.



Inherited form (example)



Final version

Once an inherited form is selected, its content appears in the current edit window. This is only a preview, it is not possible to select or modify an object in this form. To do that, you should open it in its own window.

Note You can hide the objects of an inherited form by deselecting the **Inherited form** option in the **Display** submenu in the **Form** menu or in the editor’s contextual menu.

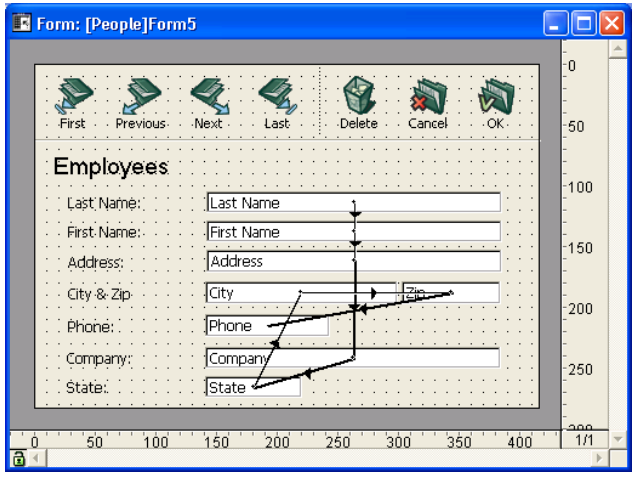
To stop inheriting a form, select the <**None**> option in the Property List.

Note It is possible to define an inherited form in a form that will be eventually used as inherited form for a third form. The combining of objects takes place in a recursive manner. 4th Dimension detects recursive loops (for example, if form [table1]form1 is defined as inherited form [table1]form1, in other words, named the same) and interrupts the form chain.

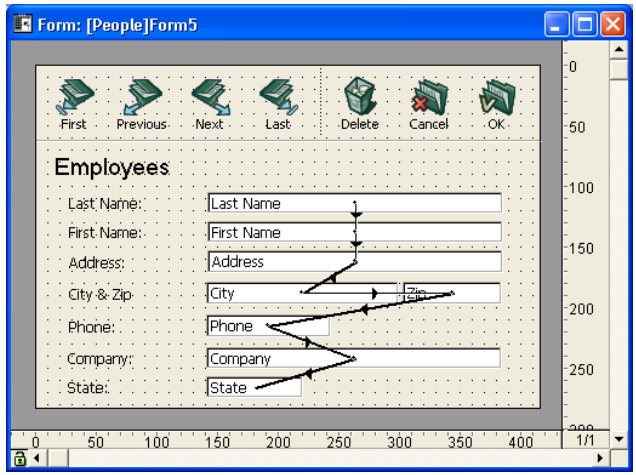
Data Entry Order

The data entry order is the order in which fields, subforms, and other tabable objects are selected as you tab through an input form. If you don't specify a custom entry order, by default 4th Dimension uses the layering of the objects to determine the entry order in the direction "background towards foreground." The standard entry order thus corresponds to the order in which the objects were created in the form.

In some forms, a custom data entry order is needed. Below, for example, additional fields related to the address have been added after the creation of the form. The resulting standard entry order thus becomes illogical and forces the user to enter the information in an awkward manner.



In cases such as this, a custom data entry order allows you to enter the information in a more logical order.



Viewing and Changing the Data Entry Order

The **Entry Order** command in the **Form** menu lets you view the current entry order of all fields in a form and allows you to create a custom entry order.

► To view or change the entry order:

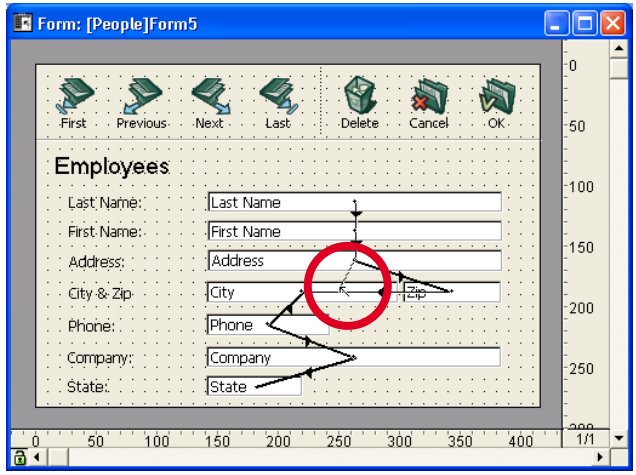
1 **Choose Entry Order from the Form menu.**

The pointer turns into an entry order pointer , and 4th Dimension draws a line in the form showing the order in which it selects objects during data entry.

Viewing and changing the data entry order are the only actions you can perform until you click any tool in the Tools palette.

2 **To change the data entry order, position the pointer on an object in the form and, while holding down the mouse button, drag the pointer to the object you want next in the data entry order.**

4th Dimension adjusts the data entry order accordingly.



4th Dimension will adjust the entry order accordingly.

3 **Repeat step 2 as many times as necessary to set the data entry order you want.**

4 **When you are satisfied with the data entry order, click any unselected tool in the Tools palette or choose Entry Order from the Form menu.**
4th Dimension returns to normal operation of the Form editor.

Note Only the entry order of the current page of the form is displayed. If the form contains enterable objects on page 0 or coming from an inherited form, the default entry order is as follows: Objects from page 0 of the inherited form ➔ Objects from page 1 of the inherited form ➔ Objects from page 0 of the open form ➔ Objects from the current page of the open form.

Setting the First Object in the Data Entry Order

All enterable objects are part of the data entry order. To define the first object of the entry order, the Entry Order mode must be disabled.

► To establish one of the objects as the first in the data entry order:

1 **Select the object you want to be first in the entry order.**

2 **Choose Move to Back from the Object menu.**
OR
Click the Move to Back tool in the Tools palette.
OR
Select Move to Back from the object contextual menu.

The object will be placed in back of any other form element. This is temporary.

3 **Choose Entry Order from the Form menu.**

The selected object becomes the first object in the entry order and the object that was first becomes second. You can now drag from object to object in order to create the data entry order you want.

4 **When you have finished, select Entry Order from the Form menu again or click any unselected tool in the Tools palette.**

The Form editor returns to normal operation.

The object you selected to make first in the entry order has returned to its previous location (it is no longer moved to the back of the form).

Using a Data Entry Group

While you are changing the data entry order, you can select a group of objects in a form so that the standard data entry order applies to the objects within the group. This allows you to easily set the data entry order on forms in which fields are separated into groups or columns.

▶ To create a data entry group:

1 **Choose Entry Order from the Form menu.**

2 **Draw a marquee around the objects you want to group for data entry.**

When you release the mouse button, the objects enclosed or touched by the rectangle follow the standard data entry order. The data entry order for the remaining objects adjusts as necessary.

Restoring the Standard Data Entry Order

You can restore the standard data entry order (based on the layering of the objects) at any time.

1 **Choose Entry Order from the Form menu.**

2 **Draw a marquee around all the objects on the form.**

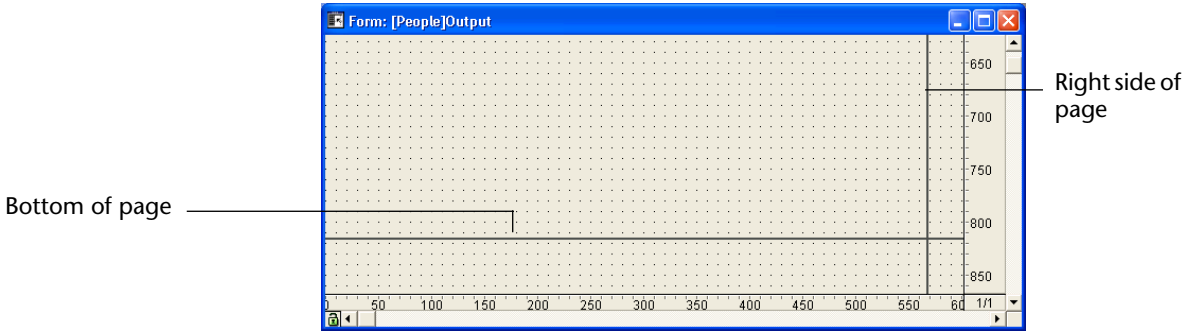
When you release the mouse button, the objects enclosed or touched by the rectangle take the standard data entry order.

Viewing and Printing Forms

Each form has an area of about 1245 square feet. You scroll to bring hidden portions of the form into view. For viewing on screen, your form design can use this entire area. You can scroll to view any element you place in the form.

For printing, form elements must fit within a single page width, but may be several pages in length. The actual size of a page depends on your printing device, the paper it is using, and the specifications you enter in the Page Setup dialog box. 4th Dimension displays page border lines in the Form editor. These lines indicate the page limits. The page border lines respond to any page setup changes. The page setup specifications are stored with the form when it is closed. The form's limits can be displayed or hidden using the **Paper** command in the **Display** submenu of the Form editor (see "Showing/Hiding Elements in the Form Editor" on page 270.

The figure below shows the page border lines:



Saving Forms

It is a good idea to save any changes you make to a form, especially when using 4D Server with multiple users. You can save a form by closing or saving it. You can close a form by clicking its Control-menu box or by choosing **Close Form:** ***Name*** from the **File** menu.

To save a form without closing it, choose **Save Form:** ***Name*** from the **File** menu.

Once a form has been saved, you can continue to work on it. If you make a mistake or do not like the changes you have made, you can revert to the last saved version of the form. This makes the form appear exactly as it did the last time that it was saved. To do this, choose **Revert to Saved** from the **File** menu.

4D Server When a form is saved in the Design environment, users are able to see your changes the next time they open the form.

6 Working with Fields and Active Objects

4th Dimension allows you to customize data entry forms so that your interface includes exactly those features that you need. You can add interface elements such as picture buttons, tab controls, drop-down lists, combo boxes, and hierarchical lists to your data entry forms. You can also implement drag and drop operations. You can use triggers, form methods, or object methods to enforce business rules during data entry.

This chapter tells you how to do the following:

- Place fields and other active objects in the form,
- Set display formats and data entry filters,
- Use data entry constraints such as maximum, minimum, default, or required values,
- Write form or object methods,
- Add interface objects such as buttons, pop-up menus or drop-down lists, combo boxes, scrollable areas, splitters and tab controls,
- Add subforms to forms,
- Attach custom menus to forms.

Active Objects Defined

An active object is anything on a form that performs a database task or an interface function. There are many kinds of active objects. Fields are considered active objects. Other active objects — enterable objects (variables), combo boxes, drop-down lists, picture buttons, and so on — store data temporarily in memory or perform some action such as opening a dialog box, printing a report, or starting a background process.

In some cases, you can specify the active object's action by making selections in the Object Properties window. For example, you can use built-in automatic button actions to specify the action of a button. In other cases, you specify the object's action by writing a method that is automatically attached to the object.

There are also instances in which you will want to manage active objects at a higher level. For example, data validation tasks can be handled by the trigger that runs automatically when a record is saved. The trigger can examine the values in each field for possible violations of business rules.

Fields in a form

The fields of a form are used to enter or display the data of a record.

When you create a new form using the Form wizard, you select the fields that you want to include in the form. Once the form has been created, you can use the Form editor to specify additional properties such as the display format and data entry controls.

These properties only apply to the forms in which they were specified. You can then use the same properties for other forms or specify new ones. You can change the properties of the fields or add/delete them at any time.

Adding Fields to a Form

You can add or delete fields from a form at any time. For example, you may decide to add fields to a form when the following occurs:

- You discover you need a field you did not choose in the Form Wizard.
- You add a field to the database structure and need to add it to a form so that you can use it.

When you place a field in a form, you can immediately set its properties. You can add a field to a form using either the Tools palette or the Explorer.

Note You can also add a field by duplicating an existing field and then modifying the duplicate's properties.

▶ To add a field to a form using the Tools palette:

1 **If the field does not exist in the database, use the Structure editor to create the field.**

For information on adding a field to a table, see the section "Creating Fields and Setting Field Properties" on page 172.

2 **Open the form to which you want to add the field.**

For information on opening a form, see the section "Opening a Form in the Form Editor" on page 288.

3 **Click the Add Field tool  in the Tools palette.**

You can either drag the tool from the Tools palette or select the tool and draw the subform area in the form (refer to "Opening a Form in the Form Editor" on page 288).

4 **Either drag and drop the tool or select the tool by clicking it and draw the field area.**

For more information about how to create an object refer to "Creating an Active Object" on page 414.

4th Dimension automatically displays the properties of the new field in either the Object Properties window or the Property List (depending on your current settings).

5 Select the field you want to insert from the table/field list.

Object Properties window

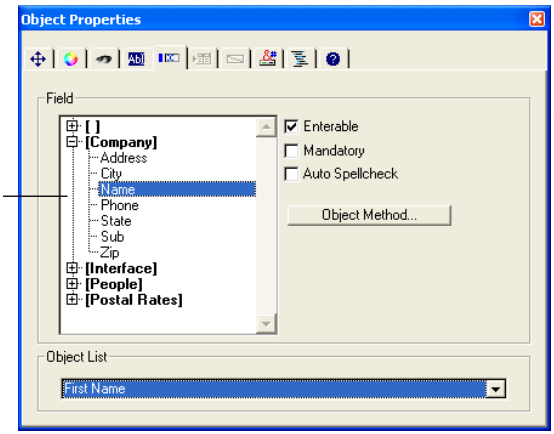


Table and Field List

Property List

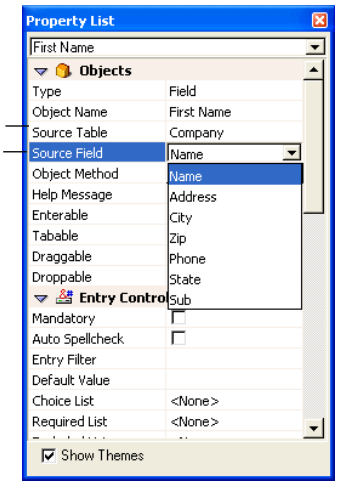


Table List
Field List

Note Fields from the master table appear twice, once at the top of the list of fields (beneath the empty brackets), and once in the table name's expanded list. If you select the field from the top list, 4th Dimension takes this to mean "field01, in this position in the master table." If you select the field from the table name's expanded list, 4th Dimension takes this to mean "this field from this table." For example, if you select the third field from the top list and then copy and paste the field into another table's form, in the new form it would become the third field in the new master table. If you select the field from the table name's expanded list and then copy and paste the field into another table's form, it would remain the same field from the same table.

6 If desired, select the specific properties you want to assign the field.

After creating the field, you generally need to set additional properties. You can set data entry controls, write help text, attach a method, set resizing or repositioning options, set platform interface, font, or appearance options. The new field appears in the form where you placed it. The field area displays the name of the field you selected.

For information on field attributes, see the sections "Field Attributes" on page 182 and "Setting the Enterable and Mandatory Attributes" on page 373.

► To add a field using the Explorer:

1 **Open the form to which you want to add the field.**

For information on opening a form, see the section "Opening a Form in the Form Editor" on page 288.

2 **Open the Explorer and click the Tables tab to display the hierarchical list of tables and fields.**

3 **Expand the table that contains the field you wish to add.**

The fields belonging to that table appear.

4 **Drag the desired field from the Explorer to the form.**

4th Dimension automatically displays the field properties in either the Object Properties window or the Property List, depending on the display settings that you selected.

After you place a field in a form, you can modify it as you would any other form object. You can resize it, change the font, choose colors for display on a color monitor, and so on. You can return to the Object Properties window or the Property List to modify the field's properties at any time.

Modifying a Field in a Form

As for any active object, you can set the field properties in the Object Properties window or in the Property List. For more information about this possibility, refer to "Displaying and Setting Form and Object Properties" on page 282.

► To modify a field's properties:

1 **Select the field you want to work with and double-click it.**

Depending on the current Display settings, either the Object Properties window or the Property List appears.

You can use the Object Properties window or the Property List to change any setting. You can even select a new field in the form to display its properties and you can select a set of fields to modify their common properties.

2 **Make any changes you like.**

Your changes take effect immediately.

Note If the field has been grouped with another object, you must ungroup it before you can display the Object Properties window.

Changing a Field into a Variable and Vice-Versa

The Property List allows you to transform every object type (active or not) into another object type. You can also transform a field into a variable and vice-versa. This is useful when, after inserting a field in a form, you want to change that field into a variable because you don't need to store the value. When 4th Dimension changes an object into another object, it keeps the original properties of the object (coordinates, object method, appearance, color and so on).

The data type assigned to a field will be kept for the variable: a picture field will be converted to a picture variable.

To change a variable into a field or a field into a variable, select the object and select Field or Variable from the **Type** drop-down list in the Property List. The Property List is then updated to display the properties for that new object type. The object name, object method, and its properties (size, enterable and so on) remain identical.

Note When you change a variable into a field, 4th Dimension assigns the first field in the first table to the object by default. You can manually define the table and field in Source table and Source field.

Inserting Dynamic Table and Field Names

You can insert dynamic table and field names into your forms (as static text). When you place dynamic table or field names in your forms, they are automatically updated throughout your database:

- either when you modify the table or field name in the Structure editor, or
- when the 4th Dimension commands Table name or Field name (in the "Structure Access" theme) are called.

This feature is particularly useful when you want to update the table or field names in the forms on-the-fly.

► To insert a dynamic table or field name in a form:

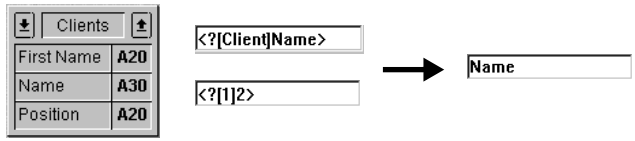
1 In a static text area, enter the following reference:

- to insert a dynamic table name: **<?[TableName]>** or **<?[2]>** (the table's creation order number, meaning the second table created).
- to insert a dynamic field name: **<?[TableName]FieldName>** or **<?[2]3>** (the table's and field's creation order number), or even **<?3>** (the field's creation order number) for the current table's field.

Please note that table and field numbers correspond to their creation order. You can add or rename tables and fields without modifying the dynamic reference system.

2 **Click outside of the text area.**

The current field's or table's name appears as it has been defined in the Structure editor.



You can view the "actual" contents of a static area in the Form editor by clicking on the area or by selecting **Show Format** or **Show Name** from the **Objects** menu.

In the User and Custom Menus environments, a table's or field's name can be modified on-the-fly by using the Table name and Field name commands. In this case, the table and field name references will display the values defined by these commands.

Note Dynamic field names is available as an option in the **Options** page of the Form Wizard. For more information, refer to "Form Display Options" on page 254.

Data Entry Controls

You can establish data entry controls for fields and enterable objects at the form level. Data entry controls restrict what the user can enter into the field or enterable object on a particular form. You can do the following:

- Set the Enterable or Mandatory attributes,
- Attach a choice list,
- Establish a list of required or excluded values,
- Set an entry filter that defines allowable characters,
- Set maximum and minimum allowable values,
- Set default values,
- Write an object method.

You can also establish data entry controls at a higher level. You have the following options:

- **Field properties** The Field Properties window (accessed from the Structure editor window) lets you set field attributes at the table level. Field properties are enforced throughout the database. In some cases, you have the option of setting a particular attribute at either the table or form level. You can set the following attributes at the table level:
  - **Mandatory** Set the Mandatory attribute for a field that is required for all records.
  - **Display Only** Set the Display Only attribute for calculated fields or other fields that are not enterable.
  - **Can't Modify** Set the Can't Modify attribute for fields that accept an initial entry but should not be changed after the record is first saved.
  - **Indexed** Set the Indexed attribute for fields on which you will be searching and sorting frequently. Also, use the Indexed attribute for primary and foreign key fields.
  - **Unique** Set the Unique attribute for the field that must be used to uniquely identify the record.
  - **Choice list** You also can associate a choice list with the field. When the choice list is assigned at the table level, it is used on all entry forms and in the Query editor.
- **Relation properties** The Relation properties window contains the Deletion control option that lets you set and enforce referential integrity. You can
  - Prevent 4th Dimension from deleting a related One record if there are related Many records,
  - Automatically delete the related Many records if the related One record is deleted,
  - Allow the user to delete a related One record even if related Many records exist (turn off referential integrity).
- **Triggers** You can create triggers that manage the process of loading, saving, and deleting records. Triggers run if a record is loaded, saved, or deleted programmatically, using any form, or during data imports and exports. With triggers, you can enforce complex business rules in a very comprehensive and systematic way. For information on using triggers, see the section "A trigger has two basic functions:" on

page 525 and the section "Triggers" in the *4th Dimension Language Reference* manual.

- **Form methods** You can create form methods that manage the use of the form.

The following sections review the data entry control options that you have at the form level. Use these options in conjunction with data entry controls that are available at the database and table levels.

Setting the Enterable and Mandatory Attributes

The Enterable and Mandatory attributes are similar to the field attributes you set in the Structure editor. If you want these attributes to be different on a particular form, you can change them here. These attributes can be set on the Field page of the Object Properties window as well as in the Property List.

These attributes do not override the field attributes set in the Structure editor. If a field already has the Display Only attribute assigned in the Structure editor, you cannot make it enterable with the Enterable form attribute. If a field already has the Mandatory attribute assigned in the Structure editor, you cannot make it non-mandatory by deselecting the Mandatory form attribute. The Enterable and Mandatory check boxes do not necessarily reflect the attribute settings in the Structure editor.

The Enterable Attribute

Every field is enterable by default. If you want to make a field non-enterable for that form, you can deselect the **Enterable** check box either in the Object Properties window or in the Property List.

A field from a related table may not be enterable if you deselected the **Enterable Related Fields** check box in the Form Wizard. You can make the related field enterable by selecting the **Enterable** check box.

For enterable objects, the **Enterable** check box is checked. You can make an enterable object non-enterable by changing the definition of the object from enterable to non-enterable by choosing non-enterable from the Type drop-down list or by unchecking the **Enterable** check box. For information about enterable objects, see the section "Enterable and Non-enterable Variables" on page 420.

Note The contents of the Property List are contextual. When the Enterable attribute is deselected in the Property List, properties that are related to entry control (Mandatory, Tabable, Entry filter and so on) disappear from the list.

The Mandatory Attribute

No field or enterable object is mandatory by default. To make the field mandatory for all forms, set the Mandatory attribute in the Field Properties window in the Structure editor. If you want to make a field or enterable object mandatory for a particular form, you can select the **Mandatory** check box in either the Object Properties window or the Property List.

Selecting the **Mandatory** check box makes a field or enterable object mandatory for that form. 4th Dimension does not accept a record if the field or object does not contain a value. For information about enterable objects, see the section "Enterable and Non-enterable Variables" on page 420.

Using Choice Lists

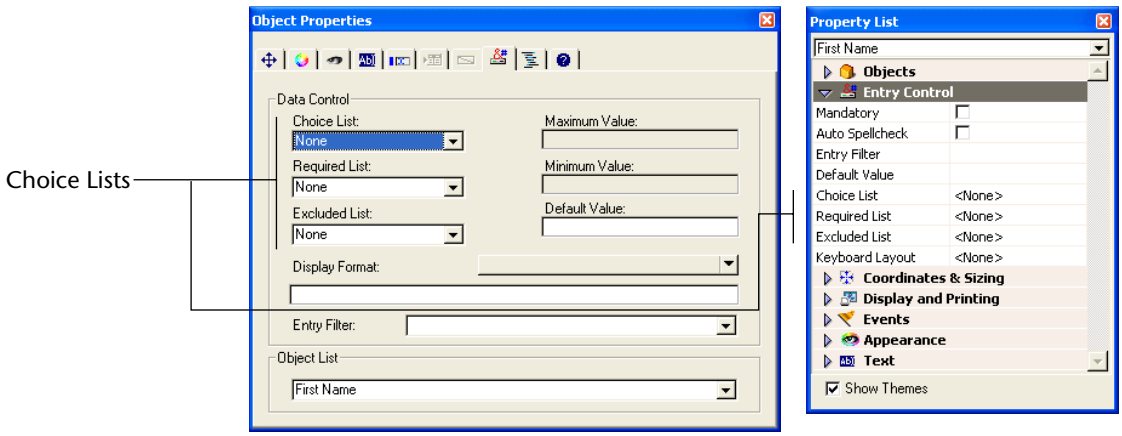
You can assign a choice list to a field at either the table or form level. If you want to assign the choice list at the table level, use the Field Properties window in the Structure editor. The choice list will then be available on all forms and in the Query editor in the User environment and in custom applications (assuming you use the Query editor in the custom application).

You can also attach a choice list at the form level. The list can serve as a choice list for that form, as a list of required entries, or as a list of excluded entries.

Choice lists can be associated with a field at the form level by either using the Data Control page of the Object Properties window or the Property List.

Object Properties window

Property List



Assigning a list at the form level gives you the freedom to vary data entry constraints from form to form. For example, a field in one input form that is used only by managers can display a comprehensive choice list while the same field in the form that is seen only by drudges has fewer choices.

If a field is already assigned a choice list at the table level, you can override the choice list at the form level. If you assign a different list to the field at the form level, then that list is used for that form only.

Before you can assign a choice list, you must have created the list in the List editor. For more information about creating choice lists, see chapter 11.

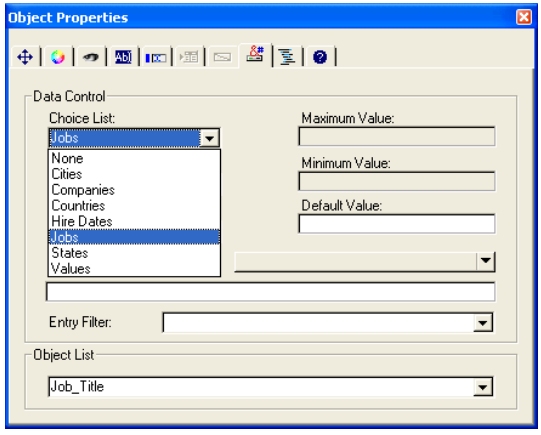
► To assign a list to a field or enterable object:

1 **Display the properties of the object in the Object Properties window or in the Property List.**

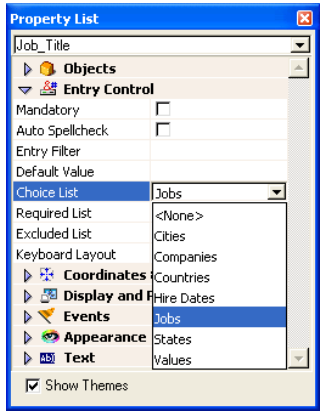
This alternative is described in the paragraph "Displaying and Setting Form and Object Properties," page 282.

2 **Choose the desired list from the appropriate drop-down list in the Data Control page of the Object Properties window or in the Entry Control theme of the Property List.**

Object Properties window



Property List



4th Dimension assigns the selected list(s) to the field or enterable object.

Choice Lists

Assigning a choice list to a field with the Choice List drop-down list causes 4th Dimension to display the choice list during data entry. The choice list appears when the field or enterable object is selected in the User environment or in custom applications. The user can then select an entry from the list. The user can overwrite the entry chosen from the choice list by typing (unless the list is also a required list).

Required Lists

A Required choice list to limit the valid entries to the items on the list. For example, you may want to require a list of job titles so that valid entries are restricted to titles that have been approved by management.

Note Making a list required does not automatically display the list when the field is selected. If you want to display the required list, assign the same list with the Choice List drop-down list.

Excluded Lists

An Excluded choice list prevents the items on the list from being entered. For example, for a field on an input form used only by drudges, you may want to attach a list of choices that can only be authorized by a manager.

Using Entry Filters

An entry filter controls exactly what the user can type during data entry. Unlike the data entry controls discussed earlier in this section, entry filters operate on a character-by-character basis. For example, if a part number always consists of two letters followed by three digits, you can use an entry filter to restrict the user to that pattern. You can even control the particular letters and numbers.

An entry filter operates only during data entry. It has no effect on data display after the user deselects the field. Usually, you use entry filters and display formats together. The filter constrains data entry and the format ensures proper display of the value after data entry. For complete information about display formats, see the section "Data Entry Controls" on page 371.

During data entry, an entry filter evaluates each character as it is typed. If the user attempts to type an invalid character (a number instead of a letter, for example), 4th Dimension simply does not accept it. The null character remains unchanged until the user types a valid character.

Entry filters can also be used to display required formatting characters so that the user need not enter them. For example, an American telephone number consists of a three-digit area code, followed by a seven-digit number that is broken up into two groups of three and four digits, respectively. A display format can be used to enclose the area code in parentheses and display a dash after the third digit of the telephone number. When such a format is used, the user does not need to enter the parenthesis or the dash.

An Introduction to Entry Filter Codes

Entry filter codes usually start with an ampersand (&). This character tells 4th Dimension to use what follows as an entry filter. If the code starts with a tilde (~), it means the same thing as "&" except that any letter is automatically made uppercase.

The & is usually followed with an "A," an "a," or a "9," meaning allow only uppercase letters (A), allow lowercase and uppercase letters (a), or allow only numbers (9). For example, &9 allows only numbers and &A allows only capital letters.

The number sign (#) tells how many digits or characters is allowed by the code. If the code uses no number signs, the filter allows as many digits or characters as you want. For example, &9 allows as many digits as is entered. The filter &9## allows only two digits.

The exclamation point (!) is sometimes used to change which character will appear on screen to indicate the number of characters the user can enter. Without an !, 4th Dimension displays an underscore for each digit or character the user can enter. For example, !?&9## displays question marks in both of the places the user will type and it allows only numbers and only two digits.

For information about creating entry filters, see the section "Entry Filter Codes" on page 380.

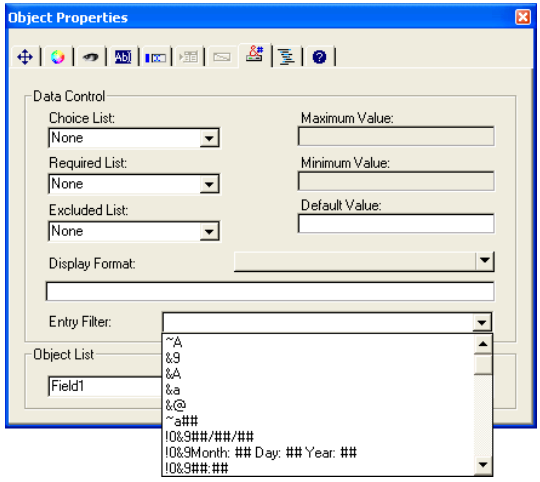
Choosing an Entry Filter

You create the entry filter by choosing a built-in or custom filter from the Entry Filter drop-down list or by typing an entry filter code into the Entry Filter Display area. The Entry Filter drop-down list contains filters for date, time, and alpha fields. The names of any custom filters you create are added to the entry filter drop-down list.

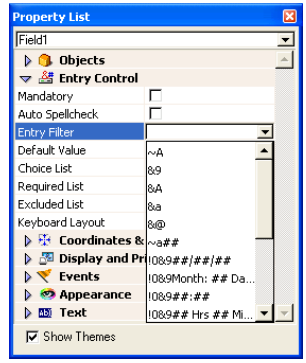
For information on creating custom filters, see the section "Creating Custom Display Formats and Entry Filters" on page 385. Most often you will find a suitable entry filter in the drop-down list.

The figure below shows an entry filter being chosen from the drop-down list.

Object Properties window



Property List



Here is a table that explains each of the entry filter choices in the Entry Filter drop-down list.

Entry Filter	Explanation
~A	Allow any letters, but convert to uppercase.
&9	Allow only numbers
&A	Allow only capital letters
&a	Allow only letters (uppercase and lowercase)
&@	Allow only alphanumeric characters. No special characters.
~a##	State name abbreviation (e.g., CA). Allow any two letters, but convert to uppercase.
!0&9##/##/##	Standard date entry format. Display zeros in entry spaces. Allow any numbers.
!0&9 Month: ## Day: ## Year: ##	Custom date entry format. Display zeros in entry spaces. Allow any number. Two entries after each word.
!0&9##:##	Time entry format. Limited to hours and minutes. Display zeros in entry spaces. Allow any four numbers, separated by a colon.

Entry Filter	Explanation
!0&9## Hrs ## Mins ## Secs	Time entry format. Display zeros in entry spaces. Allow any two numbers before each word.
!0&9Hrs: ## Mins: ## Secs: ##	Time entry format. Display zeros in entry spaces. Allow any two numbers after each word.
!0&9###-####	Local telephone number format. Display zeros in entry spaces. Allow any number. Three entries, hyphen, four entries.
!_&9(###)!0###-####	Long distance telephone number. Display underscores in first three entry spaces, zeros in remainder.
!0&9###-###-####	Long distance telephone number. Display zeros in entry spaces. Allow any number. Three entries, hyphen, three entries, hyphen, four entries.
!0&9###-##-####	Social Security number. Display zeros in entry spaces. Allow any number.
~"A-Z;0-9,;.;-"	Uppercase letters and punctuation. Allow only capital letters, numbers, space, comma, period, and hyphen.
&"a-z;0-9; ;,;.;-"	Upper and lowercase letters and punctuation. Allow only lowercase letters, numbers, space, comma, period, and hyphen.
&"0-9;.;-"	Numbers. Allow only numbers, decimal point, hyphen (minus sign).

You can modify an entry filter after you choose it from the drop-down list. For example, if you want to use a filter that allows upper and lowercase letters, but also need to allow the wildcard character (@), you could choose the filter:

&"a-z;0-9; ;,;.;-"

and change it to:

&"a-z;0-9; ;,;.;-;@"

For more information about modifying entry filters, see the section "Entry Filter Codes" on page 380.

Using Entry Filters and Display Formats Together

You often create a matching display format when you use an entry filter. An entry filter operates only during data entry. It has no effect on how the data is displayed after you tab out of the field.

For example, if you use the Social Security number entry filter (&9###-##-####), you should also choose the matching Social Security number display format (###-##-####). Without the display format, only the numbers, not the hyphens, are displayed in the field. Display formats can be used in both input and output forms and in quick reports.

Here are some suggested entry filters and matching display formats for common types of fields.

Field Type	Entry Filter	Display Format
State	~a##	(none needed)
Zip Code (standard)	&9#####	(none needed)
Zip Code (extended)	&9#####-####	#####-####
Phone number	&9###-#### &9(###) ###-#### &9 ###-###-####	###-#### (###) ###-#### ###-###-####
Social Security number	&9###-##-####	###-##-####
Date	!0&9##/##/## !0&9Month: ## Day: ## Year: ##	(Any Date Format)
Time	!0&9##:## !0&9##Hrs##Mins##Secs !0&9Hrs:##Mins:##Secs:##	(Any Time Format)

You can use display formats on input forms, output forms, and quick reports. For information about using display formats in quick reports, refer to the chapter on quick reports in the *4th Dimension User Reference*.

Entry Filter Codes

Often, you create an entry filter simply by choosing it from the entry filter drop-down list. If you need a filter for a type of field not covered by the choices in the drop-down menu, you can create a filter or modify an existing one.

This section describes how to write the code for an entry filter.

An entry filter code has three parts, in this order:

initiator "argument" placeholders.

The *initiator* informs 4th Dimension that the subsequent argument is to be used as a filter during data entry in the field. The *argument* defines the allowable characters. The *placeholders* define the places available for the characters.

For example, the following entry filter allows only the letters “a,” “b,” “c,” or “g” to be entered in two places:

&"a;b;c;g"##

In this example, the ampersand (&) is the initiator; the “a;b;c;g” is the argument; and the number signs (#) are the placeholders. The filter can be read as, “Allow the letters ‘a’, ‘b’, ‘c’, or ‘g’ in two places.” Thus the user may enter “ag,” “gc,” “ba,” “ab,” “aa,” “ac,” or any other combination of the four allowed characters.

Entry filters can be combined. The following entry filter allows only the letters “a,” “b,” “c,” or “g” to be entered in two places, followed by the numbers 1, 3, or 8 in one place:

&"a;b;c;g"##&"1;3;8"#

The user must use two of the allowed letters, followed by one of the allowed numbers.

Characters that Initiate a Filter

Two characters initiate a filter: the ampersand (&) and the tilde (~). These characters instruct 4th Dimension to use the argument that follows immediately as the filter for the subsequent placeholders.

In addition, the tilde (~) also instructs 4th Dimension to make any letters uppercase. It does not prevent a lowercase letter from being typed; it simply changes it to an uppercase letter.

The following entry filters are equivalent in their effects:

&"P"#

~"p"#

The difference between them is that the filter initiated with the ampersand (&) does not accept a lowercase p. The filter initiated with the tilde (~) accepts the lowercase p but converts it to uppercase.

Because no letters are involved, the following entry filters are equivalent:

&"1;5;8"#

~"1;5;8"#

Arguments

A filter argument follows the initiator and defines the characters that are allowed in the subsequent placeholders. To create a filter argument, surround the allowable characters with quotation marks.

Arguments are made up of lowercase letters, uppercase letters, numbers, punctuation marks, and special characters (!@#$%^&*(){}[]":';?><,./`~). If you use a lowercase letter in the argument, only the lowercase form of the letter can be typed by the user. If you use an uppercase letter in the argument, only the uppercase form of the letter can be typed by the user.

An argument may be a single character (a letter or a number), for example, “j,” “J,” or “6”.

An argument may be a set of characters separated by semicolons, for example, “a;r;t” or “1;5”.

An argument may include ranges of characters. A range is defined by the first character, a hyphen, and the last character. Examples are, “a-c” and “1-5”. The “a-c” argument is equivalent to “a;b;c”, and “1-5” is equivalent to “1;2;3;4;5”.

An argument may include single letters, single numbers, and one or more ranges, for example, “a;m-z;3;5-9”.

The following table shows useful shorthand versions of arguments. They are used in filters without quotation marks.

Character	Meaning	Equivalent
9	Allow numbers	"0-9"
a	Allow lowercase and uppercase	"a-z;A-Z"
A	Allow uppercase	"A-Z"
@	Allow alphanumeric	"a-z;A-Z;0-9"

The following entry filters are equivalent:

&9#

&"0-9"#

&"1;2;3;4;5;6;7;8;9;0"#

The following entry filters are equivalent:

&a#

&"a-z;A-Z"#

The following entry filters are equivalent:

&A#

&"A-Z"#

Placeholders

The number sign (#) is the only placeholder. You use one number sign for each character the user can enter in the field.

For example, the following entry filter allows the user to enter letters in four places:

&a####

The following entry filter allows the user to enter uppercase letters in three places, followed by numbers in two places:

&A###&9##

If you show no placeholders, the filter code allows any number of characters. The following entry filter allows the user to enter only numbers, but it does not limit the length of the entry:

&9

You can set the maximum number of characters allowed in an Alpha field in the Structure editor.

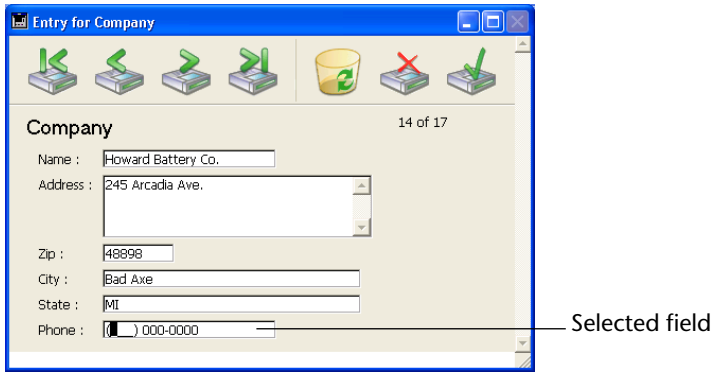
Display Characters

When a field with an entry filter is selected for data entry, 4th Dimension displays an underline (_) for each placeholder. As the user types a valid character, each underline is highlighted and replaced with the typed character.

You instruct 4th Dimension which character to substitute for the underline by beginning the entry filter with an exclamation point (!) and the character you want.

You can substitute any character for the underline. For example, if you display “XXXX” and the user types only two of the allowed characters (say they are “AA”), the field will contain “AAXX” when the record is saved.

The following illustration shows a selected field displaying underlines and zeros:

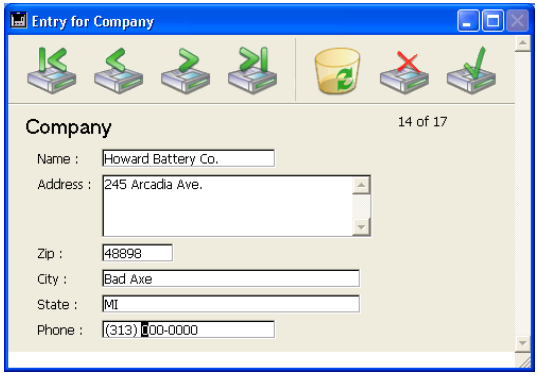


Dead Characters

Any characters, punctuation marks, and spaces can be used as dead characters. Dead characters are displayed during data entry, but they are skipped over by the insertion point and are not entered as part of the data.

The characters you want to use as dead characters are placed before, after, and between placeholders. They are displayed during data entry for clarity.

The phone number entry filter (&9(###) ###-####) uses parentheses, a space, and a dash as dead characters. After you enter a digit immediately preceding a dead character, the insertion point moves directly to the first character following the dead character. The following figure shows how the insertion point skips over the parenthesis and space after the area code to allow the user to enter the next digit in the phone number.



Custom Entry Filter Formats

You can use a custom format to enter an entry filter. All custom formats are automatically displayed on the Entry Filter drop-down list. To use a custom format as an entry filter, choose its name from the Entry Filter drop-down list or type a vertical bar followed by the format name in the Entry Filter Display area. For example, the entry:

|Part Number

installs the custom format named Part Number as the entry filter for the field.

For information about creating custom formats, see "Creating Custom Display Formats and Entry Filters" which follows immediately.

Creating Custom Display Formats and Entry Filters

You can create custom display formats and entry filters that you can refer to by name. You can use a custom format or filter name in place of the code for display formats and entry filters. Custom formats and filters are useful when you use the same display formats or entry filters in several places. If you use fields with the same entry filter in several forms, you can create the entry filter once and specify it by name wherever you need it. In addition, if you decide to change a format or filter, you need only change it in one place and it is updated wherever it is used.

You can also create display formats that correspond to the entry filters and use styles to install them as well.

You can create display formats or entry filters on the **Formatting** page ("Interface" theme) of the Preferences dialog box.

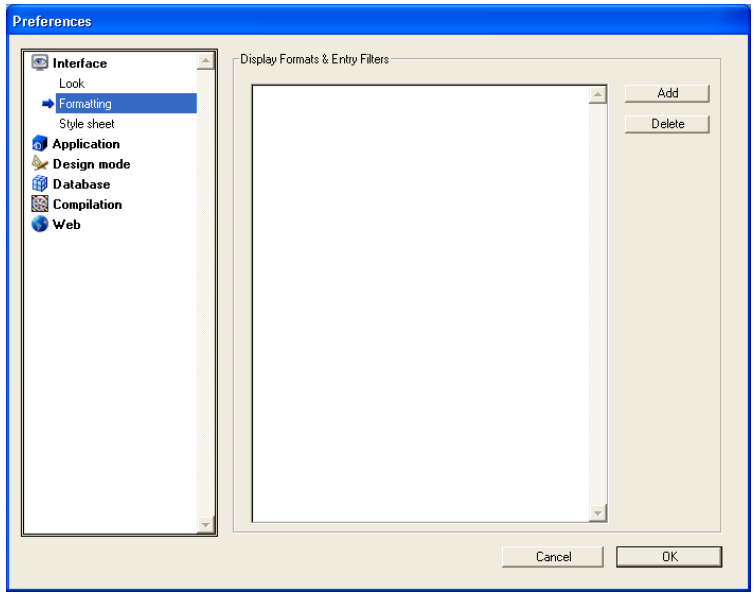
► To create a custom format or filter:

1 **Choose Preferences from the Edit menu.**
4th Dimension displays the Preferences dialog box.

For information about the Preferences dialog box, refer to chapter 2, "Setting Preferences," page 107.

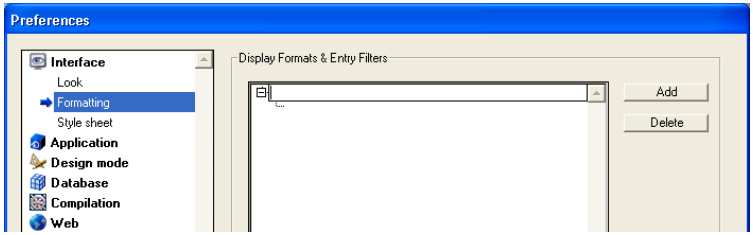
2 **Click the Formatting page under the Interface theme.**

The Formats and Filters page appears, empty by default.



3 Click Add.

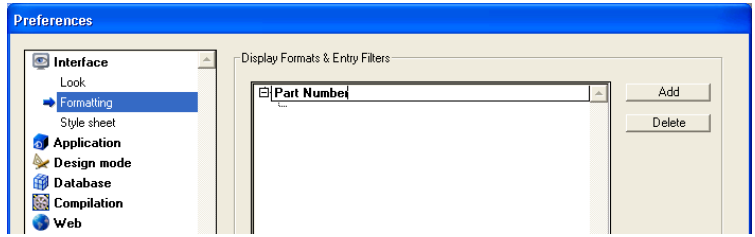
A blank item is added to the hierarchical list.



4 Type the format or filter name.

Note To edit the name of a filter or format that already exists, press the Ctrl (Windows) or Command (MacOS) key and click the name you want to edit.

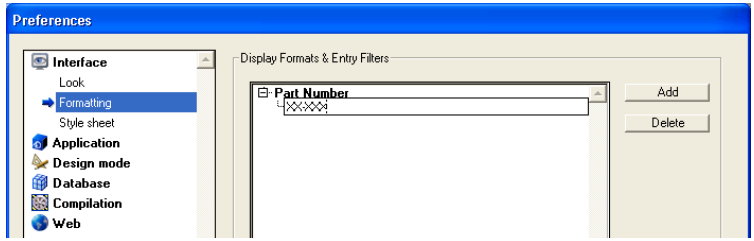
You can use up to 255 characters of any type for the name of a format or filter.



Tip You can include the word “filter” or “format” in the name to indicate its purpose.

5 **Double-click (or Ctrl+click under Windows and Command+click under MacOS) in the sub-item area below the name and type the display format or entry filter.**

You create a display filter or an entry filter just as if you were typing it into the Data Control page of the Object Properties window.



For information about creating display formats and entry filters, see the sections “Data Entry Controls” on page 371 and “Using Entry Filters” on page 376.

For example, if you wanted to create a format for a local telephone number, you would use the following:

###-####.

For example, if you wanted to create a Part Number entry filter for a part number with the format XA-654-1, you would use the following filter:

!X&"A-Z"##-!0&"0-9"###-#

and the corresponding display format is “##-###-#”.

6 **If you want to create another filter or format, click the Add button.**

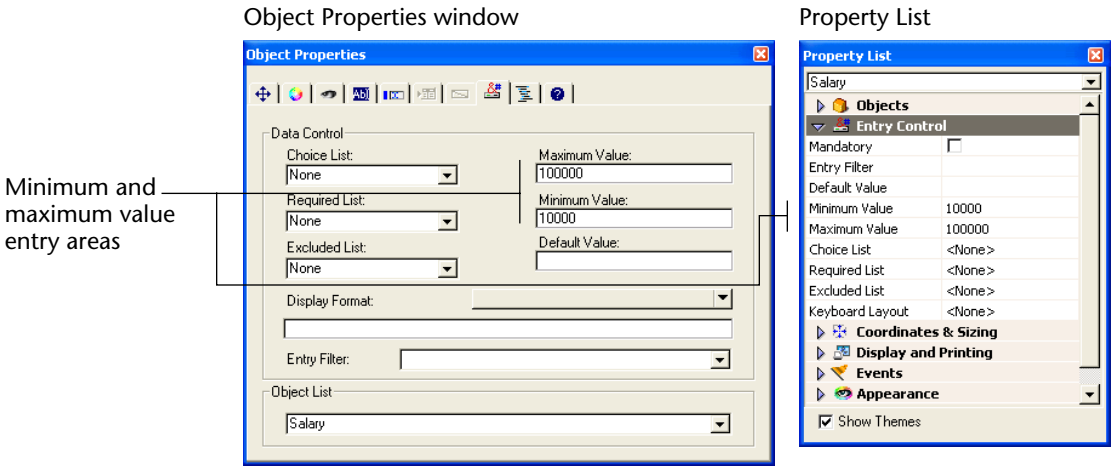
You most often create filters and formats in pairs — one for the entry filter and the other for the display format.

7 **When you have finished adding filters and formats, click on OK.**

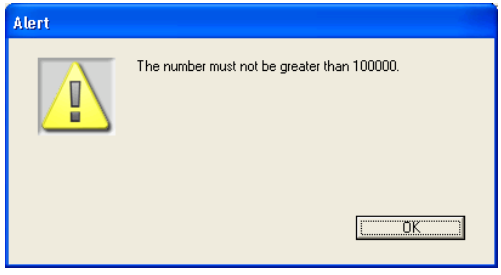
You can edit any filter or format by selecting it and changing the name or the code. You can delete any style by selecting it and clicking the **Delete** button.

Setting Maximum and Minimum Values

You can restrict a Number, Date, or Time field or enterable object by entering maximum and minimum values in the corresponding entry areas in the Data Control page of the Object Properties window.



During data entry, if the user enters a value below the minimum or above the maximum, a warning message is displayed. 4th Dimension returns the user to the field so that a valid entry can be made.



To set a maximum or minimum value, type the value you want to define the limit. Use the data entry format appropriate for the type of field or enterable object for which you are setting a limit. For example, for a Date field or object, use the date entry format to set the maximum or minimum value.

The values you set are inclusive. That is, if the user enters the same value you have set as a maximum or minimum value, the entry is allowed. Only entries lower than a minimum or higher than a maximum are disallowed. For example, if the value you set as a maximum is 15, the user can enter 15, but not 16.

4D Server Setting a maximum or minimum value changes the maximum or minimum value for all users.

You can also use methods to restrict the values that the user can enter. With a method, you can give more precise and informative feedback to the user or set minimum or maximum values based on other values in the database. For example, a method can check a customer's credit limit before validating a new transaction.

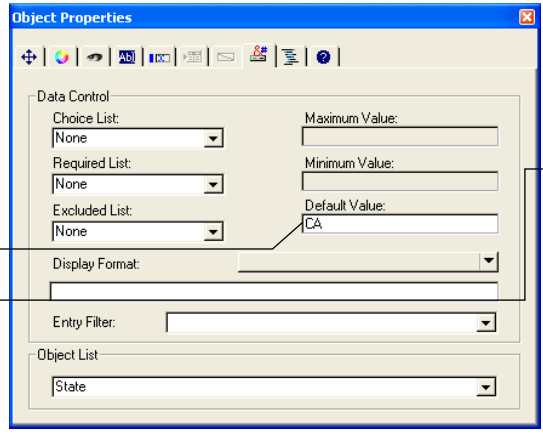
You can also use a required choice list to create unusual ranges of allowable values. For more information, see the sections "Required Lists" on page 376 and "Creating Lists" on page 646.

Setting Default Values

You can assign a default value to be entered in a field or enterable object. The default value is entered when a new record is first displayed. You can change the value unless the field or entry area has been defined as non-enterable.

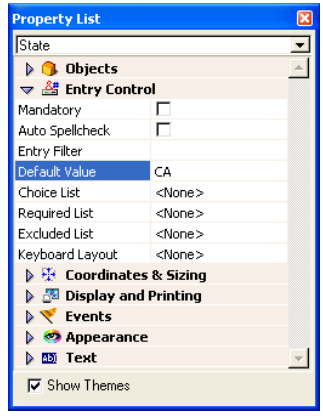
You create a default value by typing the value you want in the Default Value entry area in the Data Control page of the Object Properties window. The default value must be appropriate for the field type.

Object Properties window



Default value entry area

Property List



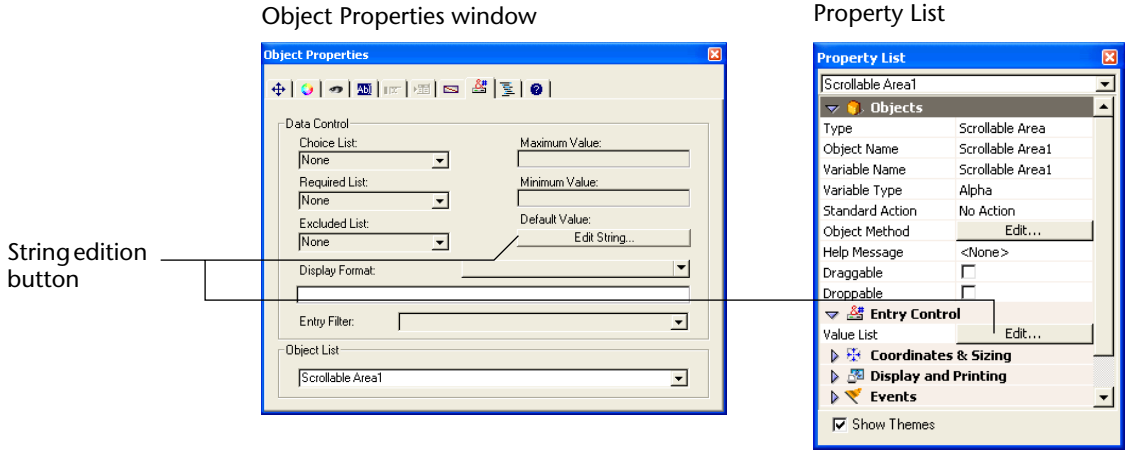
4th Dimension provides stamps for generating default values for date, time, and sequence number. The date and time are taken from the system date and time. 4th Dimension automatically generates any sequence numbers needed. The table below shows the stamp to use to generate default values automatically.

Stamp	Meaning
#D	Current date
#H	Current time
#N	Sequence number

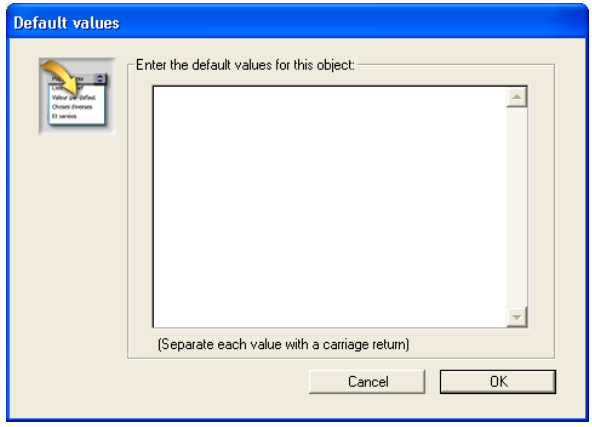
You can use a sequence number to create a unique number for each record. A sequence number is an integer (whole number) that is generated for each new record. The numbers start at one (1) and increase incrementally by one (1). A sequence number is never repeated even if the record it is assigned to is deleted from the table. Each table has its own set of sequence numbers.

Default Lists of Values

If the object displays a list of values (such as a combo box, scrollable area, pop-up menu, tab control, or drop-down list), you can specify a list of values that will be used as default values. The list will be loaded into the object prior to its being displayed on the form.
For objects that accept a list of default values, the Default Value area becomes a button:



When you click the string edition button, the Default Values dialog box appears:



Enter the list of default values. Each value should be on a separate line. Click **OK** to put away the Default Values dialog box and return to the Object Properties window.

When you enter default values into the Default Values dialog box, the values are automatically loaded into an array whose name is the name of the object. Using the language, you can manage the object by referring to that array.

4D Server Setting a default value in the Object Properties window sets the default value for all users.

Using a List to Set Default Values

If the object is a hierarchical list or a tab control, you can use a list that you created using the List editor to set default values.

► To set default values using a list:

1 **Display the object properties in either the data control page of the Object Properties window or the Property List.**

This action is described in the section "Displaying and Setting Form and Object Properties" on page 282.

2 **In the "Entry Control" section of the window, select the desired list from the Choice List drop-down list.**

Regarding Hierarchical Lists, the user can overwrite an item in the list by Ctrl+clicking (Windows) or Command+clicking (MacOS), but the changes are not saved to the actual list.

Setting Default Values Using the Language

You can also set default values using a method. For objects that accept one, you can assign the default value when the On Load event executes in the object or form method.

For objects that accept lists, you can enter the default values using the List editor and then use the Load list command in order to create a hierarchical list. You can then work with the list and its contents using the commands of the "Hierarchical Lists" theme. You can load the lists into a hierarchical list when the On Load form event occurs or load all lists in the On Startup database method.

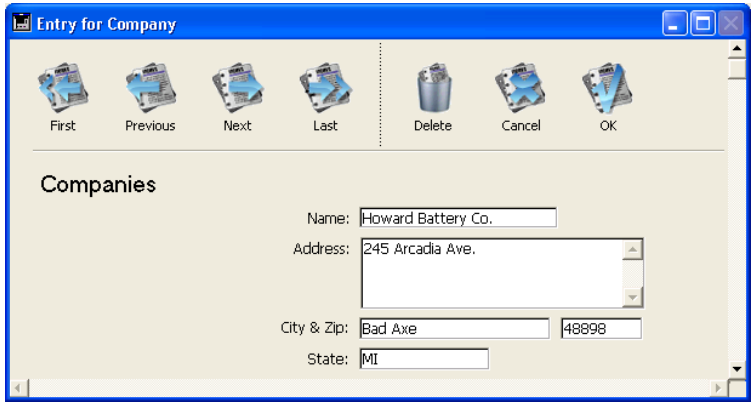
Keyboard layout

The Property List lets you assign a specific keyboard layout to each enterable field or variable in your forms. If a layout property is associated with a field or variable, the current keyboard layout of the user is automatically modified when the entry cursor is placed in the object in the User or Custom Menus environment. This function is used mainly for languages that use several alphabets (Japanese, for instance).

To associate a specific keyboard layout with a field or variable, choose a value from the **Keyboard layout** menu in the Property List ("Entry Control" theme).
When the default value (None) is used, 4th Dimension keeps the current keyboard layout.

Adding a Scroll Bar to a Text Object

Text fields and enterable objects can contain up to 32,000 characters. 4th Dimension allows you to attach a scroll bar so that the user can scroll the information. The figure below shows an input form with a text area with a scroll bar:



► To add a scroll bar to a text object:

1 **Display the object properties for the text object in either the Data Control page of the Object Properties window or the Property List.**

This action is described in the section "Displaying and Setting Form and Object Properties" on page 282.

2 **In the Object Properties window, click the Display tab**
OR
In the Property List, expand the "Display and Printing" theme.

3 **Check the Vertical Scroll Bar option.**

Note If a text field or enterable object does not have a scroll bar, the user can scroll the information using the arrow keys.

Adding Help to a Field or Object

You can add Balloon Help or a Tip to fields and active objects in your forms to help users work with your database more productively. Depending on your configuration help messages will either appear in tips or help balloons.

Note Balloon Help can be viewed only when the database is used on a Macintosh, with Balloon Help turned on. Otherwise, help messages are shown as tips which can be displayed on all operating systems.

For example, you can create a help message for a Date field which reminds the user to include a separator such as the slash mark (/) between the month, day, and year when entering data. The tip appears in the form whenever the field or object is used.

You add help to a field or object in the Help page of the Object Properties window. The help message appears only in this particular form. For information about adding Balloon Help to a field in all forms in which it appears, see the section "Choices and Help" on page 188.

The addition of existing help messages to objects can be performed in both the Object Properties dialog box or the Property List. However the creation or edition of Help messages can only be performed in the Object Properties window.

Dynamic References

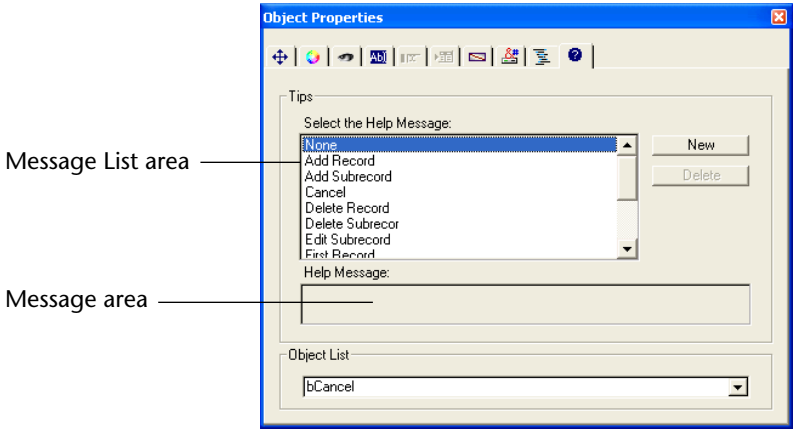
You can insert dynamic contents in a help message. The following dynamic elements can be inserted in a help message:

- a **STR# resource reference**: the syntax to apply is ":16000,2" where 16000 is the resource number and 2 is its element.
- a **table or field label**: the syntax to apply is <?[TableNum]FieldNum> or <?[TableName]FieldName>. For more information, refer to "Inserting Dynamic Table and Field Names" on page 370.
- a **variable or a field**: the syntax to apply is <VariableName> or <[TableName]FieldName. The current value of the field or variable will be displayed in the help message.
  You can, for example, enter the following text in a help message: "Enter <[Family] First_Name>'s age in this area." When in user environment, 4th Dimension will replace the field reference by the current value for the First_Name field.

► To add help message to a field or object:

1 Click the Help tab in the Object Properties window.

The Help page appears.



The list of messages contains the names of all current help messages.

2 Click New to create a new help message.
OR
Click an existing help message to edit it.

The message list is displayed in alphabetical order. The names of any new help messages are added at the beginning of the message list, before the list is sorted.

3 Enter or edit the message in the Help Message area.

The help message appears as a Balloon help on a Macintosh with Balloon Help turned on and as a Tip on any operating system.

4th Dimension stores the text for the help message so that you can reuse the message for other fields and objects.

Changing the Name of a Help Message

You can change the name of a help message found in the list of messages.

► To change the name of any help message:

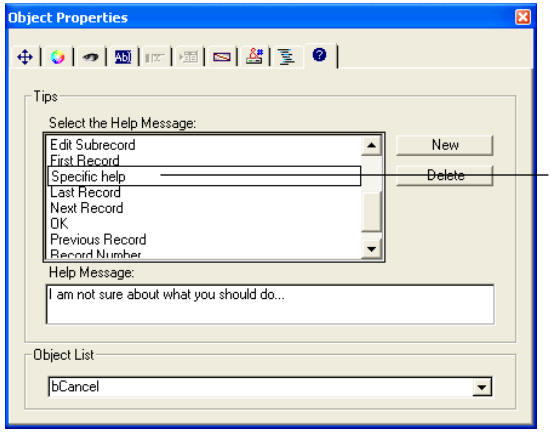
1 **Display the Object Properties window for the desired field or active object and click the Help tab.**

The Help Message page appears.

2 **Hold down the Ctrl key (Windows) or Command key (MacOS) and click the name of the message in the Message List area.**

The name of the message becomes editable.

3 **Type the new help message name then click outside of the area.**



If necessary, the list is reordered to maintain the alphabetical order.

Deleting a Help Message

To delete a help message so it no longer appears in the list of messages:

1 **Click the name of the message you want to delete in the list of messages.**

The message name is highlighted.

2 **Click the Delete button.**

The help message is removed from the list of messages.

Selecting a Help Message Once you have defined the help messages that you want to use, you can assign them to form objects using the Property List or the Object Properties window.

► To select the help message you want to use:

1 **Select the object to which you want to assign a message.**

2 **In the Help page of the Object Properties window, click the name of the message you want to select in the list of messages.**
OR
In the Property List, expand the "Objects" theme and select a message from the Help Message list.

The message you selected appears as Balloon Help and a Tip for the field or object.

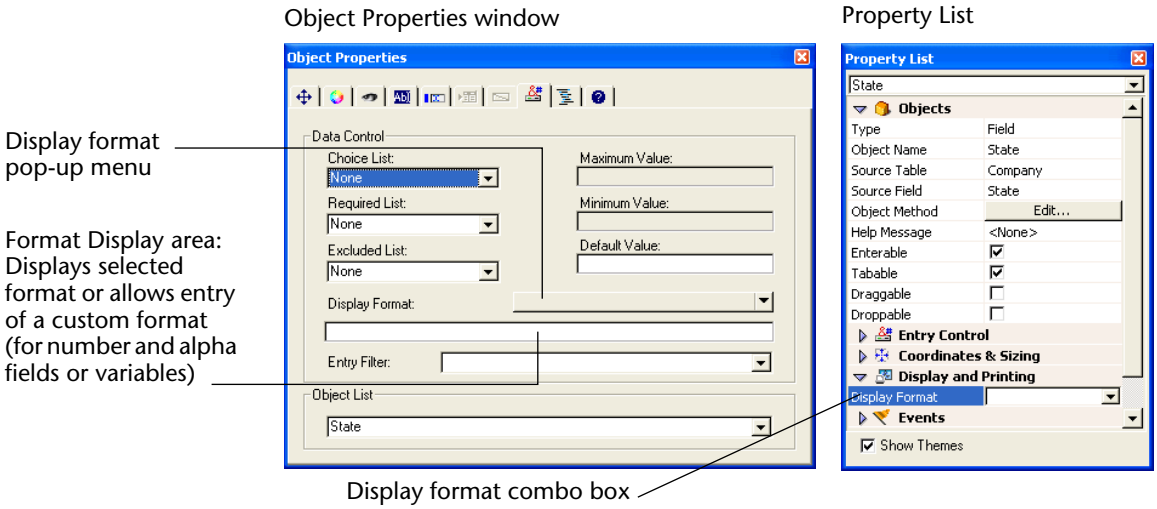
To remove the help message, select **None** in the Message list area of the Object Properties window or from the Help Message list in the Property List.

Display Formats

The display formats provided by 4th Dimension give you many choices for screen display and printing. Display formats can be applied to both fields and enterable or non-enterable areas (variables). The format you use to display the contents of a field does not affect the actual value stored by 4th Dimension.

The display format for a field can be different in each form. For example, you may want to show a value without dollar signs in an input form and display it with dollar signs in an output form.

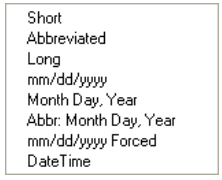
You set display formats in Data Control page of the Object Properties window.



Different formats appear in the Display Format list depending on the type of field you select. The built-in formats always appear. Any display formats that were added using the Formats and Filters editor appear in the pop-up menu along with the standard 4th Dimension formats.

Date Field Formats

Date formats control the way dates appear when displayed or printed. For data entry, you enter dates in the *MM/DD/YYYY* format, regardless of the display format you have chosen.
The figure below shows date formats in the Format pop-up menu.



Note Unlike numeric and alphanumeric formats, display formats must be selected among the 4th Dimension built-in formats.

The table below shows the Date field display formats and gives an example of each format.

Choice	Example
Short	3/25/99
Abbreviated	Wed, Mar 25, 1999
Long	Wednesday, March 25, 1999
mm/dd/yyyy	03/25/99 but 04/25/2032[1]
Month Date, Year	March 25, 1999
Abbr: Month Date, Year	Mar 25, 1999
mm/dd/yyyy Forced	03/25/1999
Date Time[2]	1999-03-25T00:00:00

1. The year is displayed using two digits when it belongs to the interval (1930;2029) otherwise it will be displayed using four digits. This is by default but it can be modified using the SET DEFAULT CENTURY command.
2. The Date Time format corresponds to the XML date and time representation standard. It is intended to be used when importing/exporting data in XML format. For more information, refer to the *User Reference* manual.

Note Regardless of the display format, if the year is entered with two digits then 4th Dimension assumes the century to be the 21st if the year belongs to the interval (00;29) and the 20th if it belongs to the interval (30;99). This is the default setting but it can be modified using the SET DEFAULT CENTURY command.

Time Field Formats

Time formats control the way times appear when displayed or printed. For data entry, you enter times in the 24-hour *HH:MM:SS* format or the 12-hour *HH:MM:SS AM/PM* format, regardless of the display format you have chosen. The figure below shows time formats in the Format pop-up menu:

The table below shows the Time field display formats and gives examples:

Choice	Example
HH:MM:SS	02:15:34
HH:MM	02:15

Choice	Example
Hour Min Sec	2 hours 15 minutes 34 seconds
Hour Min	2 hours 15 minutes
H:MM AM/PM	2:15 AM
Date Time[1]	0000-00-00T02:15:34

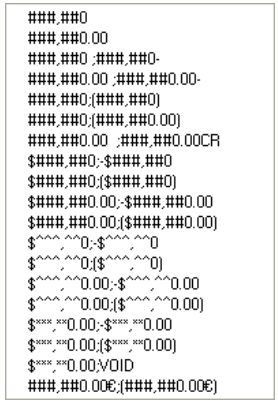
1. The Date Time format corresponds to the XML date and time representation standard. It is intended to be used when importing/exporting data in XML format. For more information, refer to the *User Reference* manual.

Note Unlike Alpha and Numeric display formats, the Time display format must be one of the formats shown in the Format pop-up menu.

Number Field Formats

Number formats control the way numbers appear when displayed or printed. For data entry, you enter only the numbers (including a decimal point or minus sign if necessary), regardless of the display format you have chosen. Number fields include Real, Integer, and Long Integer fields.

The following illustration shows the number formats in the Format pop-up menu:



You can choose the format from the pop-up menu or type it in the Format Display area. You can edit any number format in the Format Display area.

A number field can use any format including a custom format you create with the symbols you see on the Display pop-up menu. Creating a custom number display format is discussed in the following sections.

Creating a Custom Number Format

In each of the number display formats, the number sign (#), zero (0), caret (^), and asterisk (*) are used as placeholders. You create your own number formats by using one placeholder for each digit you expect to display.

For example, if you want to display three numbers, you could use the format ###. If the user enters more digits than the format allows, 4th Dimension displays <<< in the field to indicate that more digits were entered than the number of digits specified in the display format. If the user enters a negative number, the leftmost character is displayed as a minus sign (unless a negative display format has been specified). If ##0 is the format, minus 26 is displayed as –26 and minus 260 is displayed as <<< because the minus sign occupies a placeholder and there are only three placeholders.

Note No matter what the display format, 4th Dimension accepts and stores the number entered in the field. No information is lost.

Each placeholder character has a different effect on the display of leading or trailing zeros. A *leading zero* is a zero that starts a number before the decimal point; a *trailing zero* is a zero that ends a number after the decimal point.

Suppose you use the format ##0 to display three digits. If the user enters nothing in the field, the field displays 0. If the user enters 26, the field displays 26.

The table below explains the effect of each placeholder on leading and trailing zeros.

Placeholder	Effect for leading or trailing zero
#	Displays nothing
0	Displays 0
^	Displays a space[1]
*	Displays an asterisk

1. The caret (^) generates space character that occupies the same width as a digit in most fonts.

Decimal Points and Other Display Characters

You can use one decimal point in the format. If you want the decimal to display whether or not the user types it in, it must be placed between zeros.

You can use any other characters in the format. When used alone, or placed before or after placeholders, the characters always appear. For example, if you use the following format:

$##0

a dollar sign always appears because it is placed before the placeholders.

If characters are placed between placeholders, they appear only if digits are displayed on both sides. For example, if you define the format:

###,##0

a comma appears only if the user enters at least four digits.

Spaces are treated as characters in number display formats.

Formats for Positive, Negative, and Zero

A number display format can have up to three parts allowing you to specify display formats for positive, negative, and zero values. You specify the three parts by separating them with semicolons as shown below:

Positive;Negative;Zero

You do not have to specify all three parts of the format. If you use just one part, 4th Dimension uses it for all numbers, placing a minus sign in front of negative numbers. If you use two parts, 4th Dimension uses the first part for positive numbers and zero and the second part for negative numbers. If you use three parts, the first is for positive numbers, the second for negative numbers, and the third for zero.

Note The third part (zero) is not interpreted and does not accept replacement characters. If you enter ###;###;# the value zero will be displayed "#". In other words, what you actually enter is what will be displayed for the value zero.

Here is an example of a number display format that shows dollar signs and commas, places negative values in parentheses, and does not display zeros:

$###,##0.00;($###,##0.00);

Notice that the presence of the second semicolon instructs 4th Dimension to use nothing to display zero.

The following format is similar except that the absence of the second semicolon instructs 4th Dimension to use the positive number format for zero:

$###,##0.00;($###,##0.00)

In this case the display for zero would be $0.00.

Scientific Notation

If you want to display numbers in scientific notation, use the ampersand (&) followed by a number to specify the number of digits you want to display. For example, the format

&3

would display 759.62 as

7.60e+2.

Note The scientific notation format is the only format that will automatically round the displayed number. Note in the example above that the number is rounded up to 7.60e+2 instead of truncating to 7.59e+2.

Hexadecimal Formats

You can display a number in hexadecimal using the following display formats:

- **&x** This format displays hexadecimal numbers using the "0xFFFF" format.
- **&$** This format displays hexadecimal numbers using the "$FFFF" format.

Displaying a Number as a Time

You can display a number as a time (with a time format) by using "&/" followed by a digit. Time is determined by calculating the number of seconds since midnight that the value represents. The digit in the format corresponds to the order in which the time format appears in the Format drop-down menu.

For example, the format:

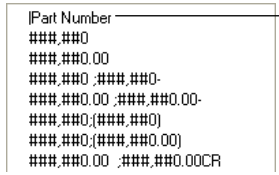
&/5

corresponds to the 5th time format in the pop-up menu, specifically the AM/PM time. A number field with this format would display 25000 as:

6:56 AM.

Custom Number Formats

You can use custom number display formats, created in the Preferences window. The custom format (as well as filter) names appear at the beginning of the alpha and number format lists, preceded by a vertical bar (|).:



For more information about creating custom formats, refer to "Creating Custom Display Formats and Entry Filters" on page 385.

Examples

The following table shows how different formats affect the display of numbers. The three columns — Positive, Negative, and Zero — each show how 1,234.50, –1,234.50, and 0 would be displayed.

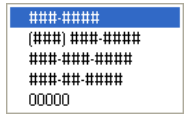
Format Entered	Positive	Negative	Zero
###	<<<	<<<	
####	1234	<<<<	
#######	1234	–1234	
#####.##	1234.5	–1234.5	
####0.00	1234.50	–1234.50	0.00
#####0	1234	–1234	0
+#####0;–#####0;0	+1234	–1234	0
#####0DB;#####0CR; 0	1234DB	1234CR	0
#####0 ;(#####0)	1234	(1234)	0
###,##0	1,234	–1,234	0
##,##0.00	1,234.50	–1,234.50	0.00
^^^^^^^	1234	–1234	
^^^^^^0	1234	–1234	0
^^^,^^0	1,234	–1,234	0
^^,^^0.00	1,234.50	–1,234.50	0.00
*******	***1234	**–1234	*******
******0	***1234	**–1234	******0
***,**0	**1,234	*–1,234	******0
,0.00	*1,234.50	–1,234.50	*****0.00
$*,**0.00;–$*,**0.00	$1,234.50	–$1,234.50	$****0.00

Format Entered	Positive	Negative	Zero
$^^^^0	$ 1234	$–1234	$ 0
$^^^0;–$^^^0	$1234	–$1234	$ 0
$^^^0 ;($^^^0)	$1234	($1234)	$ 0
$^,^^0.00 ;($^,^^0.00)	$1,234.50	($1,234.50)	$ 0.00
&2	1.2e+3	-1.2e+3	0.0e+0
&5	1.23450e+3	-1.23450e+3	0.00000

Alpha Field Formats

Alpha formats control the way the alphanumeric characters appear when displayed or printed. You can choose a format from the Format list or type a display format into the Format Display area. After choosing a format from the pop-up menu, you can edit it in the Format Display area.

The figure below shows an alpha format being chosen from the format pop-up menu.



The Format pop-up menu contains formats for some of the most common alpha fields that require formats: US telephone numbers (local and long distance), Social Security numbers, and zip codes.

The following table shows the Alpha field formats and the types of fields they are typically used for.

Format	Field type
###-####	Telephone number (local)
(###) ###-####	Telephone number (long distance)
###-###-####	Telephone number (long distance)
###-##-####	Social Security number
00000	Zip Code

You can also enter and edit a format in the Format Display area. You can edit a format after choosing it from the Format pop-up menu, but you can edit a style only in the Styles editor. You can also select the custom format name specified in the **Formatting** page (“Interface” theme) of the Preferences dialog box. In this case, the format cannot be modified in the Object properties.

The number sign (#) is the placeholder for an Alpha field display format. You can include the appropriate dashes, hyphens, spaces, and any other punctuation marks that you want to display. You use the actual punctuation marks you want and the number sign for each character you want to display.

For example, consider a part number with a format such as:

RB-1762-1.

The alpha format would be:

##-####-#.

When the user enters "RB17621," the field displays:

RB-1762-1.

The field actually contains "RB17621."

If the user enters more characters than the format allows, 4th Dimension displays the last characters. For example, if the format is:

(###)

and the user enters "HAPPY," the field displays:

(PPY).

The field actually contains "HAPPY." 4th Dimension accepts and stores the entire entry no matter what the display format. No information is lost.

Custom Alpha Formats

You can use a custom format for Alpha fields. Any custom formats or filters that you have created using the **Formatting** page of the Preferences dialog box are automatically added to the beginning of the alpha and number format lists. You can choose a custom format just as you would choose a built-in format.

For information about creating custom formats, see the section "Creating Custom Display Formats and Entry Filters" on page 385.

Boolean Field Formats

Boolean fields can contain one of two values: TRUE or FALSE. A Boolean field can be displayed as either a pair of radio buttons or as a check box.

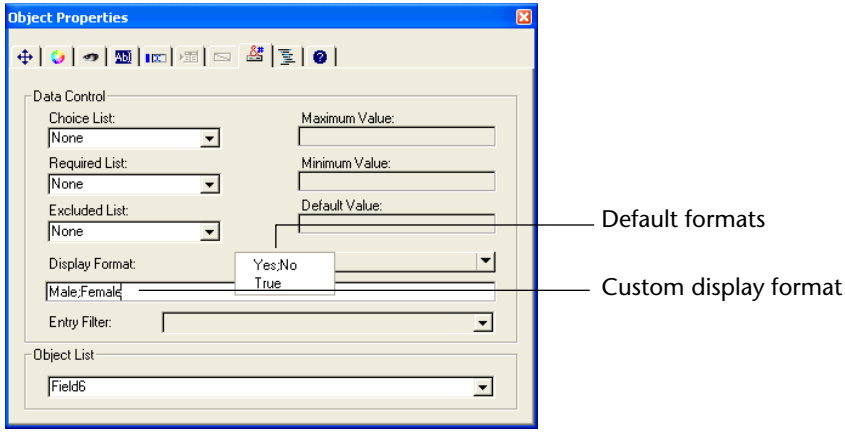
If you do not create a display format for a Boolean field, 4th Dimension automatically displays the field as a pair of radio buttons labeled Yes and No.

You can specify the form and label(s) of a Boolean field in the field properties. If you want to display only the buttons or check boxes and not the field name, you can delete the field label in the form.

Formatting a Boolean Field as Radio Buttons

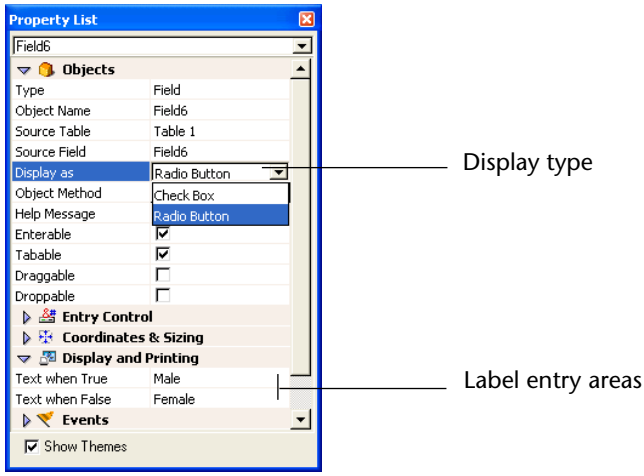
You can define a format for a Boolean field using the Object Properties window or the Property List.

- Using the Object Properties window:
  Entering two labels separated by a semicolon (;) in the Format Display area. For example, a field for sex could have one button named Male and one button named Female. To create these buttons, you would type, "Male;Female."



- Using the Property List:
  In the Property List, the appearance of Boolean fields is set using the **Display as** list, located in the Objects theme. Once you have selected **Radio Button** from that list, you can enter the labels for each value in

the **Text when True** and **Text when False** entry areas under the "Display and Printing" theme.



The buttons are displayed in the Form editor side by side as shown below.

If you use labels with different first letters, you can select the radio button by typing the first letter during data entry. For example, you can press "M" to select Male or "F" to select Female when the field is selected.

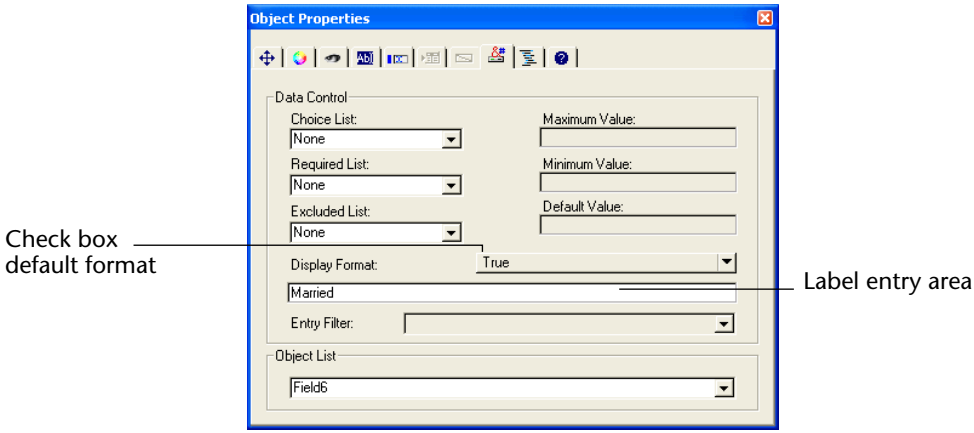
The following rules apply when the field is being used for data storage: if the first button is selected, the field is true; if the second button is selected, the field is false. The field is false by default.

Formatting a Boolean Field as a Check Box

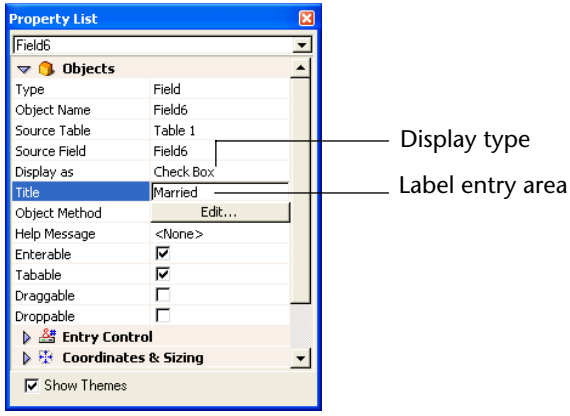
You can set a Boolean field to be displayed as a check box using either the Object Properties window or in the Property List.

- Using the Object Properties window
  Enter a single label in the Format Display area. For example, a field for Paid could have one check box labeled Yes. To create this check box you would enter "Yes." To create a check box with no label, enter a space in the Format Display area.

The following illustration shows how a Boolean field is formatted as a check box.



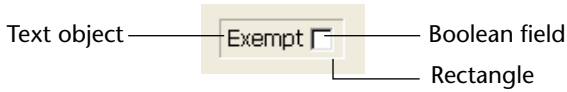
- Using the Property List
  Choose **Check box** from the **Display as** list located in the Objects theme. Once you have selected that option, an entry area labeled Title is displayed below the list. This is the entry area in which you enter the label of the check box. The default label is the field's name.



The following rules apply when the field is being used for data storage: if the check box is selected, the field is True; if the check box is deselected, the field is False. The field is False by default.

In the User environment, this field is displayed as a check box:

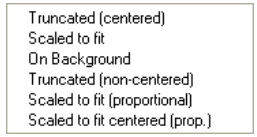
You can format a Boolean field as a check box with no label by entering a space as the display format in the Object Properties window or in the Title area in the Property List and setting a transparent border line style. In this case, you add the label for the check box as a separate object. You can then place the label wherever you want, draw a rectangle and insert dynamic references in the label (refer to "Working with Text Areas" on page 336). In the following example, the Boolean field is on top of the text object.



Picture Field Formats

Picture field formats control how pictures appear when displayed or printed. For data entry, the user always enters pictures by pasting them from the Clipboard or by drag and drop, regardless of the display format.

The figure below shows the picture formats available in the Picture Format list of the Property List ("Display and Printing" theme) or in the Display Format list of the Object Properties window.



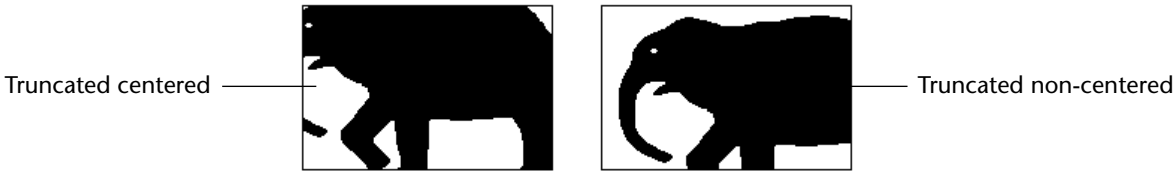
The truncation and scaling options do not affect the picture itself. The contents of a Picture field is always saved. Only the display on the particular form is affected by the picture display format.

Truncated (Centered and Non-centered)

The Truncated (Centered) format causes 4th Dimension to center the picture in the field and crop any portion that does not fit within the field area. 4th Dimension crops equally from each edge and from the top and bottom.

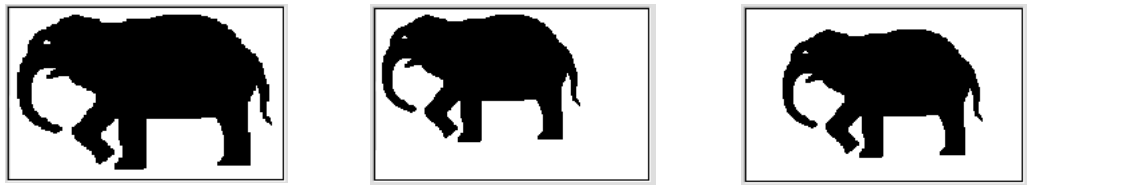
The Truncated (Non-centered) format causes 4th Dimension to place the upper-left corner of the picture in the upper-left corner of the field and crop any portion that does not fit within the field area.

4th Dimension crops from the right and bottom. The figure below compares the Truncated Centered and Non-centered formats.



Scaled to Fit

The Scaled to Fit formats cause 4th Dimension to resize the picture to fit the dimensions of the field area. The figure below compares the three Scaled to Fit formats.



When you use Scaled to Fit (Proportional), the picture is reduced proportionally on all sides to fit the area created for the picture. The Scaled to Fit Centered (Proportional) option does the same, but centers the image in the picture area.

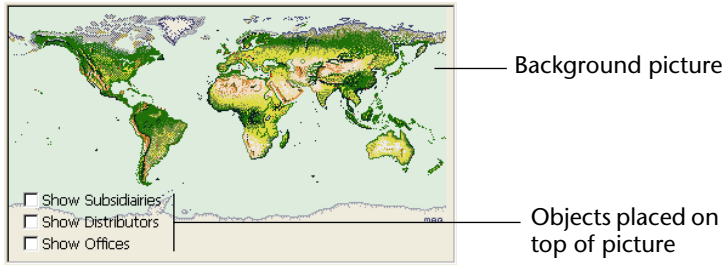
If the picture is smaller than the area defined in the form, the picture will not be modified.

If the picture is bigger than the area defined in the form, the picture is proportionally reduced. Since the picture is proportionally reduced, it will not appear distorted.

On Background

On Background makes the picture transparent. Any objects placed behind the graphic such as fields or variables are visible through the graphic. When a Picture field is in this format, the user can move the picture around the inside of the Picture field by dragging it. 4th Dimension remembers the object's position on the background.

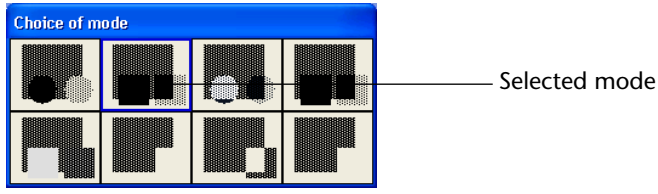
The figure below shows a form that includes a picture with the On Background format.



Note If you are printing pictures with the On Background format, they will be printed as bitmaps.

Setting the Choice of Display Mode

When a picture is displayed using the On Background setting, you can select the mode that specifies the interaction between the background and foreground colors. To display the dialog box that allows you to select the desired mode, double-click the picture area in the User environment:



The currently selected color mode is outlined by a blue line. To select a different mode, click it.

If your intent is to use the picture only as a background object for the form (not as data), you can instead paste the picture into the Picture Library and then add the picture to the form by dragging. For more information, see “Placing a Picture from the Picture Library” on page 345.

Active Objects on a Form

This section describes the active objects you can use on a form to control database and interface functions. Active objects include buttons, radio buttons, check boxes, tab controls, pop-up menus, drop-down lists, hierarchical pop-up menus and hierarchical lists, combo boxes, scrollable areas, splitters, gauges, and plug-ins. Enterable objects (variables) are treated much the same as fields and are described in the previous section.

This section explains how to create and modify active objects other than fields and enterable objects. The different active object types are discussed in detail in the section, "Types of Active Objects" on page 419.

Setting Object Properties

An object's properties can be modified using the Object Properties window or using the Property List. These two windows allow you to set numerous options.

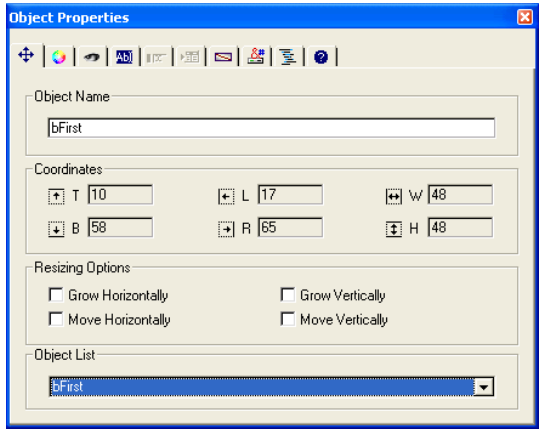
► To modify the properties of an active object:

1 **Select the object that you want to modify and double-click it.**

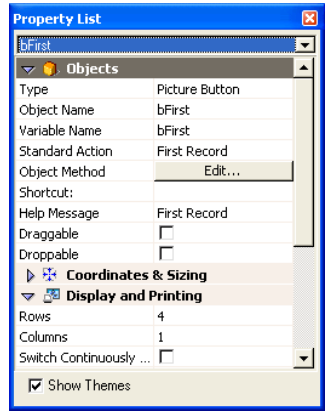
Depending on the current display settings, the Object Properties window or the Property List is displayed.

You can use either window to change the object's properties. You can also select another object to display its properties. You can also select several objects to edit their common properties.

Object Properties window



Property List



Note If the object is grouped with other objects, you first need to ungroup it before you can display its properties.

2 Apply your modifications.

Your modifications take effect immediately.

You can name an object, specify its type, define its action, drag and drop properties, resizing or repositioning options, platform interface and appearance, and attach a method for the object.

The Object Properties window and the Property List allow you to set the following properties:

- **Object** and **Variable** Each active object is associated with a variable. The variable name can be different from the object's name. When using 4th Dimension's language, you can refer to an active form object by its variable name or object name.

  You can set the object type in both the Property List and the Object Properties window but the variable type can be set only in the Property List.

  You can also assign a standard action to the object. For more complex actions you can write an object method.

- **Attributes** Each object is assigned standard attributes that define how it behaves on the form. This includes, for instance, drag and drop management and the display of the focus. Some objects such as pictures may also have specific attributes.

- **Entry control** For each enterable object you can define entry controls such as entry filters, list of acceptable or unacceptable values, and so on.

- **Coordinates** You can set the size, location, the resizing options, and the positioning options for each object.

- **Colors and Appearance** You can individually set the foreground and background colors, the style, the platform interface, and the appearance of object.

- **Font** For each object that includes some text or a label, you can define the font it uses as well as its alignment. You can also use a style to set font attributes.

- **Subform** These are considered as active form objects. They have a specific set of properties.

- **Events** You can set a list of events for which the object's method will be executed.
- **Help messages** As with fields, you can assign a tip or help balloon to any active object in a form.

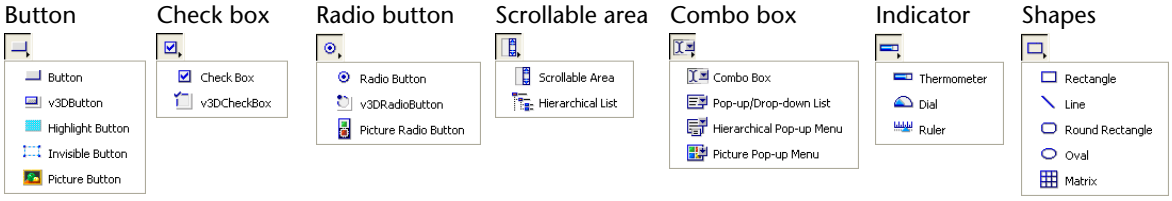
Creating an Active Object

Active objects are created using the Form editor's Tools palette. When using the Tools palette to create an active object you can do it by drawing the object with the tool or by dragging and dropping the tool itself.

The Active objects tools are located in the upper part of the Tools palette.

Using this palette you can create numerous types of active objects: text variables, tab controls, any type of button, check boxes, radio buttons, pop-up menus and drop-down lists, hierarchical menus, combo boxes, scrollable areas, indicators, picture menus, button grids, splitters, and plug-in areas.

Some objects are grouped by types; the subtypes appear in drop-down lists associated with a button in the Tools palette. These drop-down lists are shown below.



For more information about the contents and the use of the Tools palette, refer to "The Tools Palette" on page 273. For a detailed description of the types of active objects, please refer to "Types of Active Objects" on page 419.

► To create an active object in a form:

1 **Open the form in which you want to create the active object.**

For more information on how to open a form, refer to "Opening a Form in the Form Editor" on page 288.

2 **(Optional) If the desired object type is in a drop-down list, first choose the object from the drop-down list.**

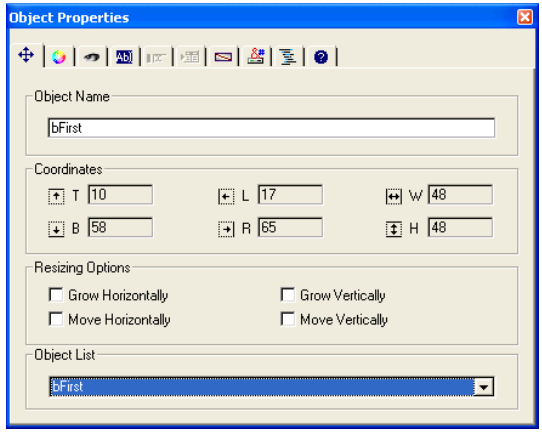
3 **Select the tool that corresponds to the object you want to create and either drag it onto the form or click it and draw an area on the form.**

For more information, refer to "Creating Objects" on page 312.

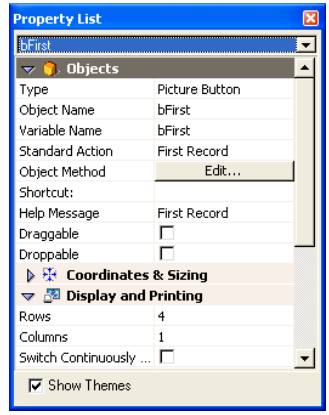
4th Dimension automatically displays the properties of the new object in the Property List or in the Object Properties window (depending on the current settings).

4 **Define the object by entering its name or editing the default name.**

Object Properties window



Property List



The object's name is used in methods when you refer to that object, so it is necessary to make sure that this name is unique.

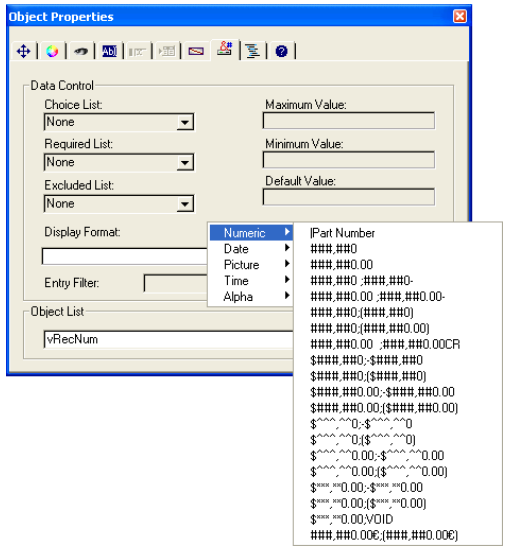
5 **Set the object's properties to match your requirements.**

For more information about the creation of objects of a particular type, refer to "Types of Active Objects" on page 419.

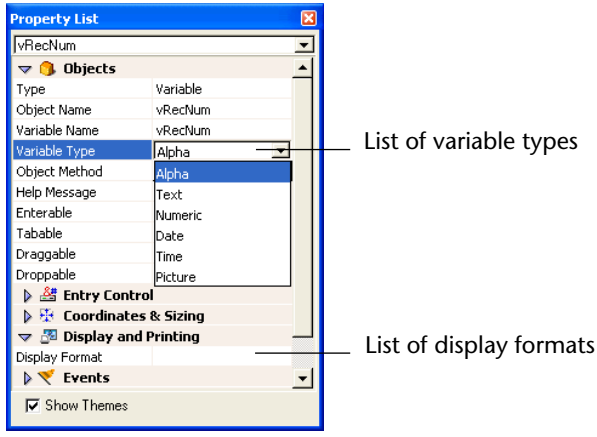
Display Formats for Objects

Both the Object Properties window and the Property List provides the same display formats for both enterable and non-enterable objects as it does for fields. The difference is that 4th Dimension does not know what data type is to be displayed or used in the object.

- In the Object Properties window, the Display format pop-up menu is a hierarchical pop-up menu. You choose the data type (Numeric, Alpha, Date, Time, or Picture) and then the display format.



- In the Property List, there is an additional option that lets you select the type of data that will be processed by the variable. By default this type is Alpha. Once a value is selected the Display format list is updated to reflect the formats available for this type of variable:



The selected format is displayed in the Display Format area. If you choose a numeric or alpha object type, you can edit the format in the usual way. You can type formats for objects directly into the Display Format area. For more information about display formats, see the section "Display Formats" on page 396.

Data Entry Controls for Enterable Objects

The Object Properties window provides data entry controls for enterable objects as well as for fields. These controls allow you to:

- Set an entry filter that defines allowable characters,
- Display a choice list,
- Establish lists of required values or excluded values,
- Set maximum and minimum allowable values,
- Set default values or default lists of values.

These controls work for enterable objects exactly as they work for fields. For complete information, see the appropriate sections earlier in this chapter.

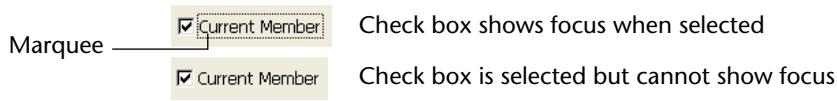
Setting the Tabable and Focusable Properties

The Tabable and Focusable properties can be set in the Display page of the Object Properties window or in the Objects theme of the Property List. Those two options will affect how the object appears during data entry.

- When the Focusable property is selected for an object, the object is outlined by a gray dotted line when it is selected.
- If an object has the Tabable property, it is included in the data entry order. That is, the user can press the Tab key to select the object.

The Tabable property is only accessible if the Focusable property is selected. That is, any tabable object shows the focus when it is selected. However, some objects can be "focusable" while not being "tabable" (for example, an object can be selected by clicking it and not "tabable"). In this case, the object does not belong to the entry sequence.

For example, the following illustration shows the effect of the Focusable property on check boxes.



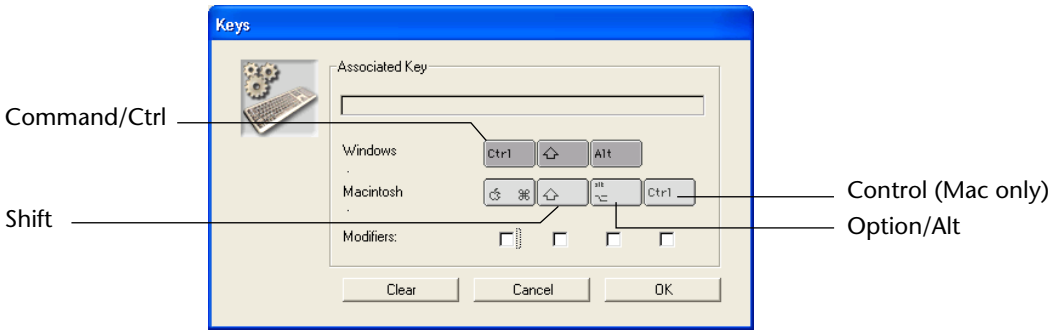
Note A non-enterable variable cannot be "tabable."

Assigning a Keyboard Shortcut

The Variable page of the Object Properties window allows you to assign a keyboard shortcut for buttons and check boxes. The user can then activate the button or select the check box using the keyboard instead of having to use the mouse.

► To assign a keyboard shortcut:

1 **Click the Keys button in the Variables page of the Object Properties window.**
OR
In the Property List, click the [...] button in the Shortcuts property located in the "Objects" theme[1].
The Shortcuts dialog box appears.



2 **Type the keyboard shortcut.**
For example, if you want to use Ctrl+h, hold down the **Ctrl** key and press h. The letter h will then appear in the Associated Key area and the check box below the Ctrl key will be checked.
You are not required to use modifier keys. You can use any key alone as the shortcut, although this is not recommended in most cases.
If you like, you can manually modify the selection of modifier keys by selecting or deselecting any of the modifier key check boxes.
To start over again, click **Clear**.

3 **When you have finished, click OK.**
The Property List displays the keyboard shortcut that was assigned to the object.
If you want to change the shortcut later, simply open the Shortcuts dialog box and type the key combination you want to use.

1. You may need to click in the Shortcut entry area in the Property List to make the button visible.

4th Dimension displays the new combination in the Shortcuts dialog box.

Enabling Drag and Drop

Active objects on a form may have drag and drop properties. Two options located in the Display page of the Object Properties window and in the Objects theme of the Property List, are related to drag and drop.

- **Dragable** controls whether the user has the ability to drag the object,
- **Dropable** controls whether the object has the ability to "receive" a dragged object.

  If you want to enable drag and/or drop for a particular object, enable the appropriate property. You then must manage the drag and drop action using a method. For more information, see the section on Drag and Drop commands in the *4th Dimension Language Reference* manual.

Types of Active Objects

4th Dimension provides the following types of active objects (in addition to fields):

- Text variables (enterable and non-enterable)
- Button
- 3D Button
- Highlight button
- Invisible button
- Picture button
- Check box
- 3D Check box
- Radio button
- 3D Radio button
- Picture radio button
- Scrollable area
- Hierarchical list

- Combo box
- Pop-up menu/Drop-down list
- Hierarchical menu
- Picture pop-up menu
- Tab control
- Plug-in area
- Button grid
- Thermometer
- Indicator
- Ruler
- Splitters.

The sections that follow describe each kind of object in detail.

Enterable and Non-enterable Variables

An enterable variable allows the user to enter a value into a variable and display the value. A non-enterable variable allows you to display the value of a variable.You use methods to manage enterable and non-enterable variables.

Variables are used for temporary storage of data. One common use for a variable is to display calculations that are done using a method such as:

```
vTotal:= Quantity * Price
```

You create a variable that displays the result of the calculation, name the object vTotal, and use a method to do the calculation.

An enterable variable accepts data. You can set data entry controls for the object as you would for a field. The entered data is associated with the variable name. You can manage the data with object or form methods using the object's name as a variable.

A non-enterable variable only displays data. The displayed data is associated with the object's name. You control the data with methods, using the object's name as a variable. The form events On Clicked and On Double Clicked can also be used with non-enterable objects. This function particularly makes managing customized contextual menus easier.

To create a non-enterable variable, create a standard variable and deselect the **Enterable** property. This property can be set in the Variable page of the Object Properties window and in the Options theme of the Property List.

Enterable and non-enterable variables can be of any size. Since they display characters, when the object area is resized, it snaps to a size depending on the object's font size. Variables that contain alphanumeric characters, numbers, dates, times, and pictures can make use of display formats. Text objects can use a scroll bar and can be printed with variable frame. Picture variables can be scaled or truncated.

Buttons

The Form editor lets you add a wide variety of buttons to your forms. When you add buttons to a form, you can associate a standard action to each button. Automatic buttons let the user accept, cancel, or delete records, move between records, move from page to page in a multi-page form, and open, delete, or add records in a subform.

You normally add buttons when you create the form using the Form Wizard. You can modify these buttons' actions in the Object Properties window or the Property List. For example, you can remove a standard action from a button and write an object method that specifies the button's action.

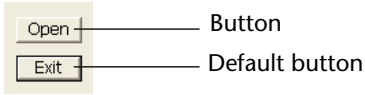
You can also add buttons and assign button actions with the Form editor. For example, if you need more than one subform on the form, you can add the necessary additional subforms and automatic buttons in the Form editor. You simply add each button to the form and associate the standard action with each button.

You can associate automatic button actions to the following types of buttons:

- **Buttons** These are standard text buttons that are displayed as boxes. Button text is displayed in the selected font, font size, style, and color.

When you create a button, you should type a label into the Button Text area. The label appears inside the button when the form is used in the User environment or custom applications. You can change the text that appears in the button at any time by returning to the Object Properties window and modifying the text in the Button text area.

- **Default Buttons** A default button looks exactly like a button, except that is border is thicker. This indicates to the user that the button is the recommended choice. The following illustration compares a button to a default button.



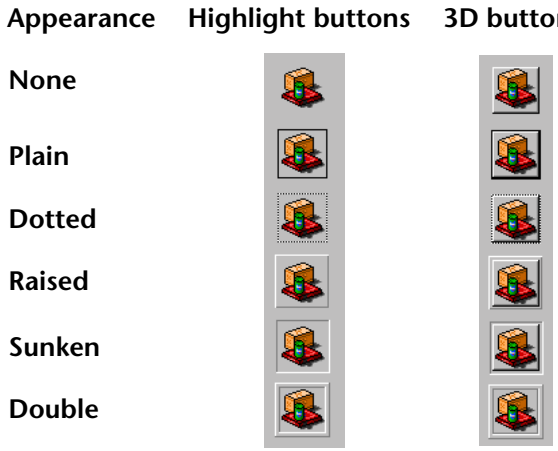
Under MacOS X, the default buttons are pulsing buttons if the Mac Theme platform appearance is selected:

Notes
- There can only be one Default button per form page.
- In the Property List, the Default button object type does not exist. It is a property that you can select for standard buttons.

- **Highlight buttons and 3D buttons** These buttons are designed to be placed on top of graphic objects. They are invisible until clicked. When the user clicks a highlight button, the button is highlighted. Highlight buttons are dimmed in appropriate circumstances in the User environment and in custom applications.

  The appearance of highlight and 3D buttons depends on the Appearance settings of the form on which the button appears and the button's Appearance settings.

  The diagram below illustrates the variations:

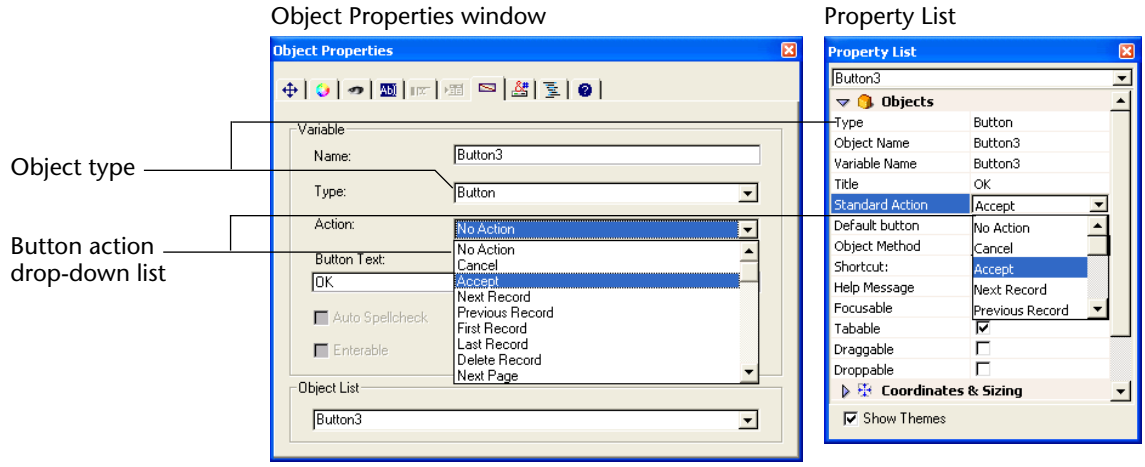


- **Invisible buttons** These buttons are invisible and do not highlight when clicked. The resulting action, such as displaying a different page, should indicate that the button has been clicked. An invisible button should be placed on top of text or a graphic that denotes its function; the user clicks on the text or graphic, and the button is activated.

Button activation management

Buttons are dimmed when appropriate in the User or Custom Menus environments. For example, if the first record of a table were displayed, the First Record button would appear dimmed.

You create a button by choosing the desired button type from the Type drop-down list. You then choose the automatic button action you want from the Action drop-down list.



Regardless of the type of button, you must select an item from this drop-down list. If you want the button to perform an action not listed in the Action drop-down list, choose “No Action” and write a method that specifies the button’s action. Normally, you would activate the On Clicked event in the Events page and the method would run only when the button is clicked.

Note All variables associated with buttons (including regular buttons, highlight buttons, invisible buttons, radio buttons, radio pictures, or check boxes) are initialized to 0 when the form is first opened in the User environment. When the user clicks a button, it becomes 1. You can associate a method with any button.

Standard Button Actions This section discusses each standard action that can be assigned to a button.

- **No Action** Use a **No Action** button for a button that does not perform a standard action. Choose No Action when you need to write a method to manage the button. For example, a button that displays a custom Find dialog box in a custom application would have a No Action standard action because you must write a method to open the custom dialog box.
- **Accept and Cancel Actions** Clicking an **Accept** button saves a record. It triggers an On Validate event. Clicking a **Cancel** button exits the current record without saving any changes.
- **Delete Record Action** Clicking a **Delete Record** button displays an alert asking the user to confirm the deletion. Clicking Yes in the alert deletes the current record. If the user is using the input form for a subrecord, **Delete Record** deletes the current subrecord.

  After the user clicks a **Delete Record** button, 4th Dimension automatically returns to the output display.

Note A button with this action is automatically disabled when a new record is being added.

- **Record Navigation Actions** The **Next Record**, **Previous Record**, **First Record**, and **Last Record** buttons first accept the current record and then make the specified record current. The specific record made current by these buttons depends on the sort order.

  These buttons perform the appropriate actions for subrecords when the user is entering subrecords.

  A button of this type is automatically disabled when its action is inappropriate. For example, if the user displays the first record, the **Previous Record** button would be disabled.
- **Page Navigation Actions** **The First Page**, **Last Page**, **Next Page**, and **Previous Page** buttons display the appropriate page in a multi-page form. If there is only one page, these buttons are inactive.

  A button of this type is automatically disabled when its action is inappropriate. For example, if the user displays the first page, the **Previous Page** button would be disabled.

- **Subform Actions** The **Subform** buttons affect records in subforms. You can open, delete, or add to records displayed in subforms. Placed on a parent record's form, they affect subrecords or related records in a subform. The following are the Subform button actions:
  - **Open Subform** This button is active when a record in a subform is selected. If the user selects a record and clicks the **Open Subform** button, the Full Page form for that record opens and the user can modify the record.
  - **Delete Subform** This button is active when a record has been selected in a subform. The **Delete Subform** button does not display an alert, but erases the record immediately.
  - **Add to Subform** This button is active when a subform has been selected. When the user clicks an **Add To Subform** button, 4th Dimension creates a new record in the related table or subtable, scrolls to the record, and places the insertion point in the first enterable field in the subform.

  For more information about using the Subform buttons, refer to "Adding a Subform to the Form" on page 468.
- **Automatic Splitter** This standard action allows you to create custom splitters on a form.

  This action can only be assigned to an invisible button. When an invisible button is assigned this action, it behaves in the same way as a a splitter. By pasting a picture in the invisible button, you can create any custom interface for splitters.

  For more information about splitters, refer to "Splitters" on page 456.
- **Undo:** cancels the last action performed (=**Undo** command of the **Edit** menu). *Undo* should not be confused with *Cancel* (= cancels any modifications made to a record during its viewing and returns to the Output form).
- **Cut**: removes the selection and places it in the Clipboard.
- **Copy**: places a copy of the selection in the Clipboard.
- **Paste**: inserts the contents of the Clipboard at the location of the insertion point.
- **Clear**: deletes the selection. If nothing is selected, it erases the entire area containing the cursor (enterable areas only).
- **Select All**: selects all of the selectable elements in the context.

- **Show Clipboard**: opens a new window that displays the current contents of the Clipboard.
- **Design**: brings the windows and menu bars of the 4th Dimension Design environment to the foreground.
- **User**: brings the windows and menu bars of the 4th Dimension User environment to the foreground.
- **Custom Menus**: brings the windows and menu bars of the 4th Dimension Custom Menus environment to the foreground.
- **Quit**: displays an "Are you sure?" confirmation dialog box then exits the 4D application if validation occurs. Otherwise, the operation is cancelled.
  When this action is assigned to a button with which an object method is also associated, the following sequence is executed: first, the confirmation dialog box appears. If it is validated, 4D executes the object method. After its execution, the application quits.

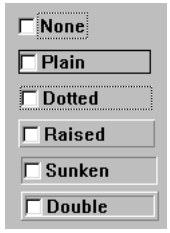
Note MacOS X Under MacOS X, the Custom Menus command associated with the **Quit** action is automatically placed in the application system menu, when the database is running in this environment. This mechanism simplifies the management of the **Quit** command under MacOS X.

Check Boxes and 3D Check Boxes

A check box is used to enter or display binary (true-false) data. It is a type of button. A check box is either selected or deselected. Its effect is controlled by a method. Like all buttons, a check box is initialized to 0 when the form is first opened. The method associated with a check box executes when the check box is selected.

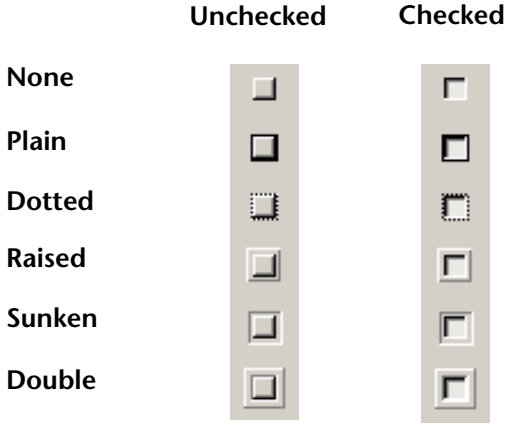
A check box displays text next to a small square. When the user selects the check box, the square is checked (an "X" is placed in it). When a check box is selected, it has the value 1. When deselected, it has the value 0. Any or all check boxes in a form can be selected or deselected. A group of check boxes allows the user to select several items.

The following illustration shows the effect of each Appearance setting on check boxes.



A 3D check box is similar to a highlight button. You place it on top of a graphic that indicates the function of the check box. When a 3D check box is clicked, its appearance changes according to the Appearance settings in the Display page of the Object Properties window. A 3D check box retains its state (0 or 1) until the user clicks it again.

The following illustration shows the effect of each Appearance setting.



Unlike a Boolean field that is formatted as a check box, the values of the check box variable are not stored automatically. You use a method to manage the variable.

Radio Buttons, 3D Radio Buttons, and Radio Pictures

Radio buttons, 3D radio buttons, and radio pictures are objects that allow the user to select one of a group of buttons or pictures. A radio button shows a small bull's-eye and text. 3D radio buttons and radio pictures display an icon or picture. They are placed on top of a graphic.

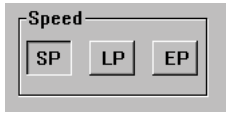
Each type of radio button is selected the same way — you click the object to select it. You can also click a selected radio picture to deselect it, but you cannot do this with a radio button.

3D radio buttons and radio pictures are similar to a highlight button in that it is transparent until selected. When selected, it highlights the picture behind it until another radio picture is selected.

The remainder of this section uses the term “radio button” to mean any type of radio button.

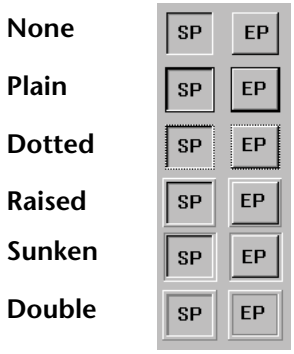
Radio buttons exist in exclusive sets. The name of each object in a group of radio buttons must begin with the same letter (e.g., bRadio1, bRadio2, bRadio3). The effects of radio buttons are controlled with methods. Like all buttons, a radio button is initialized to 0 when the form is first opened. A method associated with a radio button executes when it is selected.

The following is an example of a group of 3D radio buttons used in a video collection database to enter the speed of the recording (SP, LP, or EP).



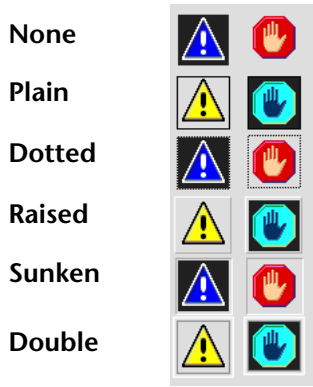
The labels are provided by separate Text objects that are placed on top of each 3D radio button. The “None” Appearance setting was used in the above example.

The following illustration shows the effects of each Appearance setting. In each case, the SP tape speed is selected.



Selecting one radio button in a group sets that button to 1 and all the others in the group to 0. Only one radio button can be selected at a time.

The following illustration shows the effects of each Appearance setting on radio pictures. The selected radio picture is black on white.



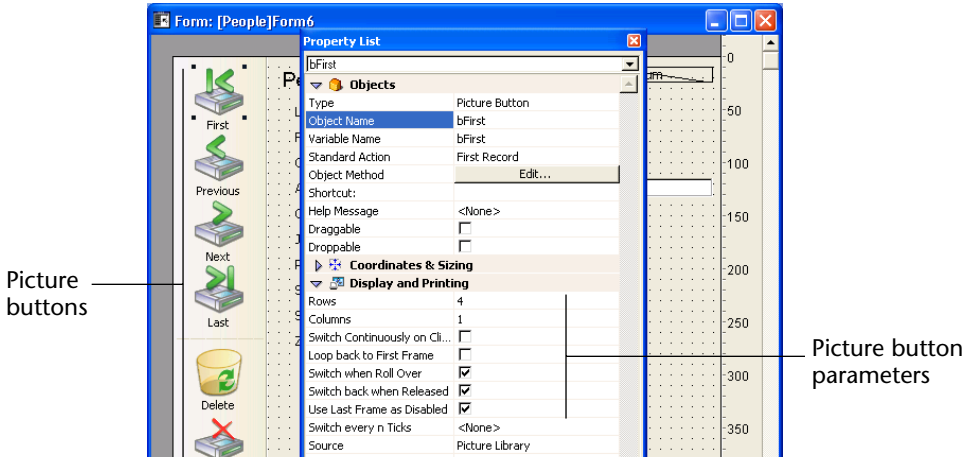
3D radio buttons retain their state (0 or 1) until another radio button in the set is clicked.

Picture Buttons

A picture button can have several different states— by comparison, a standard button accepts three states: enabled, disabled and clicked. As the name indicates, with a picture button each state is represented by a different picture.

Picture buttons can be used in two ways:

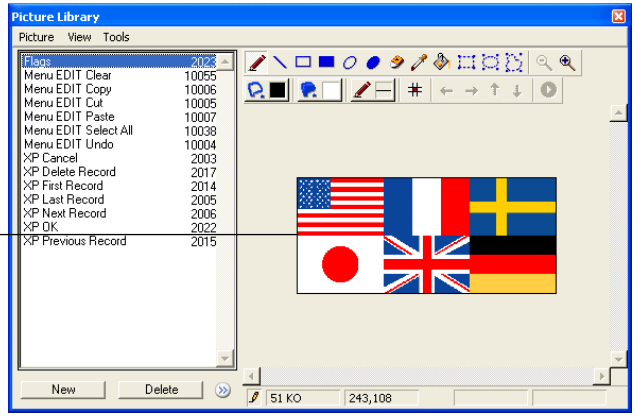
- as command buttons in a form. In this case, the picture button generally includes four different states: enabled, disabled, clicked and rolled over.
  This is the setting used by the Form wizard for most form templates:



- as a picture button letting the user choose among several choices. In this case, a picture button can be used in place of a pop-up picture menu. With a pop-up picture menu, all choices are displayed simultaneously (as the items in the pop-up menu), while the picture button displays the choices consecutively (as the user clicks the button).
Here is an example of a picture button. Suppose you want to give the users of a custom application the opportunity to choose the interface language for the application. You implement the option as a picture button in a custom properties dialog box, as shown below.

You implement the picture button in the following manner. First, you prepare *one* graphic in which the series of pictures is arranged in a row, column, or a row by column grid.
You can add this graphic to the Picture Library, to a picture variable or to a PICT resource (under MacOS).

The following illustration shows the graphic in the Picture Library.



You can organize pictures as columns, rows, or a row by column grid (as shown above). When organizing pictures as a grid, they are numbered from the left to the right, row by row. For example, the second image of the second row of a grid that consists of four rows and three columns, is numbered 5.

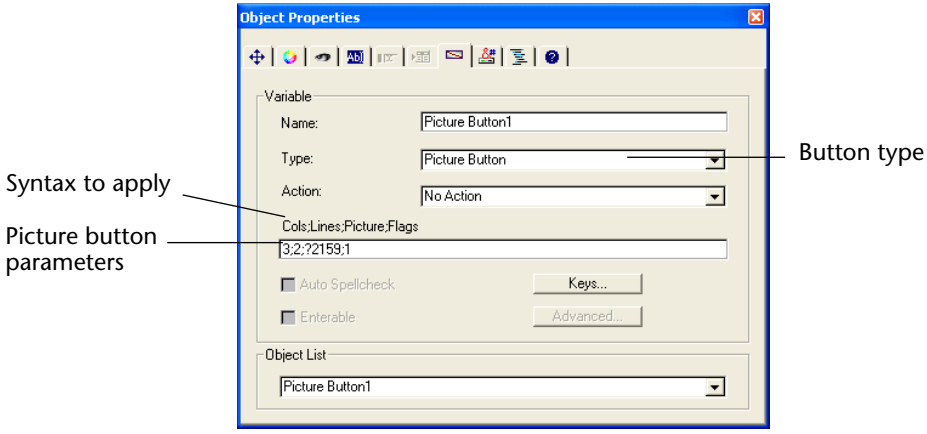
Note The Picture Library includes features that allow you to organize a picture as a table of thumbnails. It also allows you to preview the effects of the current settings.

When a picture has been defined as a table of thumbnails, you can create a picture button by dragging the picture name from the library. For more information on the Picture Library, refer to Chapter 12, "Using the Picture Library" on page 659.

The procedure for setting the properties of a picture button in the Form editor depends on whether you use the Object Properties window or the Property List. For more information about this alternative, refer to the paragraph "Displaying and Setting Form and Object Properties," page 282

Using the Object Properties window

In the Variable page of the Object Properties window, choose Picture Button as the object type and set the properties as follows.

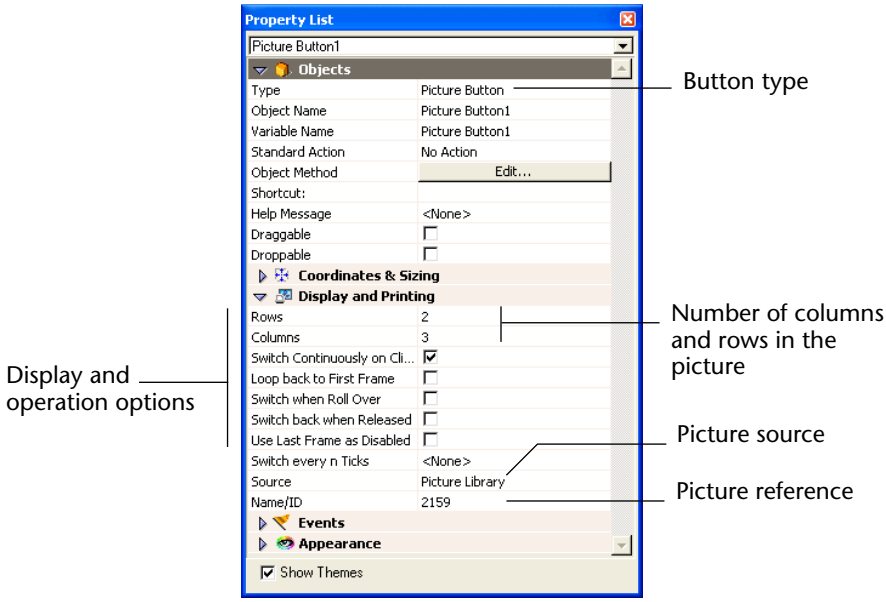


- *Cols* and *Lines* give the dimensions of the array of pictures. Since this example arranges the pictures in a grid, the specification is 3 columns and 2 lines.
- *Picture* identifies the picture in the Picture Library or in the resource file. If the picture is in the Picture Library, precede its number with a question mark "?"; if the picture is a resource, precede the resource ID with a colon ":"; if the picture is a variable, enter the variable name.

- *Flags* controls the appearance of the object and the behavior of the object when the user clicks on the object. *Flags* is the sum of any of the following values: 0, 1, 2, 4, 8, 16, 32, 64 and 128. Each of these values represents a display mode or an operation mode. For instance, if you want to enable the modes 1 and 64, enter 65 as flag value. For more information about the display modes, "Display Modes and Picture Button Operation" on page 433.

Using the Property List

In the Property List, select the **Picture Button** type in the Objects theme.



- In the **Display and Printing** theme, enter the number of columns and rows of the picture.
- In the **Display and Printing** theme, select the source of the picture from the source list. The sources that are available for selection are Variable, Picture Library, or resource. Once the source is set, enter in the **Name/ID** area
  - the name of the variable if the source is a variable, or
  - the resource number or the picture number if the source is a resource of the Picture Library.

- The remaining options allow you to set the display mode as well as the operation of picture buttons. The organization of these options is different from that of the Object Properties window and is described in the next section.

Display Modes and Picture Button Operation

The following table describes the display and operation modes you can set for picture buttons. In the Object Properties window you do this by setting a value for the Flag property and in the Property List you choose the property shown in the second column of the following table.

Value for Flag	Option in the Property List	Description
0	None	Displays the next picture in the series when the user clicks; it displays the previous picture in the series when the user holds down the **Shift** key and clicks. When the user reaches the last picture in the series, the picture does not change when the user clicks again. That is, it does *not* cycle back to the first picture in the series
1	Switch Continuously	Is similar to previous except that the user can hold down the mouse button to display the pictures continuously (i.e., as an animation). When the user reaches the last picture, the object does *not* cycle back to the first picture.
2	Loop Back to First Frame	Is similar to previous except that the pictures are displayed in a continuous loop. When the user reaches the last picture and clicks again, the first picture appears, and so forth.
16	Switch when Roll Over	The contents of the picture button are modified when the mouse cursor passes over it. The initial picture is re-established when the cursor leaves the button's area. This mode is frequently used in multimedia applications or in HTML documents. The picture that is then displayed is the last picture of the thumbnail table, unless the Use Last Frame as Disabled option is selected (128). If that option is selected, it is the next-to-last thumbnail that is displayed.
32	Switch Back when Released	This mode operates with two pictures. It displays the first picture all the time except when the user clicks the button. In that case, the second picture is displayed until the mouse button is released, whereupon it switches back to the first picture. This mode allows you to create an action button that displays its status (idle and clicked). You can use this mode to create a 3D effect or display any picture that depicts the action.

Value for Flag	Option in the Property List	Description
64	Transparent (Appearance theme)	Controls transparency. If you use 64, the picture button background is transparent. The effect of transparency is shown in the following illustration: 
128	Use Last Frame as Disabled	This mode allows you to set the last thumbnail as the thumbnail to display when the button is disabled. When this mode is selected, 4th Dimension displays the last thumbnail when the button is disabled. When this mode is used in addition to the modes 0, 1, and 2, the last thumbnail is not taken into account in the sequence of the other modes. It will appear only when the button is disabled.
Additional parameter located after Flags	Switch every n Ticks	This mode allows you to cycle through the contents of the picture button at the specified speed. In the Object Properties window, this mode is selected by entering a fifth parameter in the button's syntax. Adding this parameter tells 4th Dimension that the picture button will cycle through the thumbnails every n ticks, where n is the parameter. For example, if you enter "2;3;?16807;0;10" in the Object Properties window, the picture button will display a different picture every ten ticks. In the Property List, you set this parameter by entering a value in the Switch every n Ticks line. When this mode is active, only the Transparent mode can be used (64).

For example, you want to set a button that accepts the following modes: Switch back when Released, Switch when Roll Over and Use Last Frame as Disabled.

In the case of a table of thumbnail that has one row of four columns, each thumbnail correspond to the following statuses: default, clicked, roll over and disabled.

In the Property List you would set the following properties: 1 row, 4 columns as well as the options **Switch back when Released**, **Switch when Roll Over** and **Use Last Frame as Disabled**.

In the Object Properties window, you would enter the following syntax: “4;1;?15000;176”.

Pop-up Menus, Drop-down Lists, and Scrollable Areas

Pop-up menus, drop-down lists, and scrollable areas are objects that allow the user to select from a list. You manage the items displayed in the pop-up menu or scrollable area using an array.

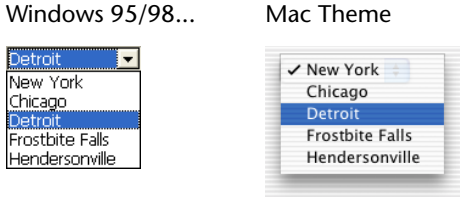
An array is a list of values in memory that is referenced by the name of the array.

A pop-up menu/drop-down list displays an array as a list of values when you click on it.

Note Under MacOS, a drop-down list is called a pop-up menu. These two objects are functionally identical.

A scrollable area displays the array in a list box that can be scrolled and used to select an item.

The figure below shows the same drop-down list/pop-up menu with two different appearances:



A scrollable area displays the array in a list box that can be scrolled and used to select an item.

You initialize the object by loading a list of values into an array. You can do this in several ways:

- In the Data Control page of the Object Properties window, click the **Edit Strings** button to enter the list into the Default Values dialog box.

In the Property List, click the Edit button located next to the value List line in the Entry Control theme. For more information, see "Default Lists of Values" on page 390. The default values are loaded into an array automatically. You can refer to the array using the name of the object.

- Before the object is displayed, execute code that assigns values to the array elements. For example:

```
ARRAY TEXT (aCities;6)
aCities{1}:="Philadelphia"
aCities{2}:="Pittsburg"
aCities{3}:="Grand Blanc"
aCities{4}:="Bad Axe"
aCities{5}:="Frostbite Falls"
aCities{6}:="Green Bay"
```

This code could be placed in the form method and be executed when the On Load form event runs.

- Before the object is displayed, load the values of a list into the array using the LIST TO ARRAY command. For example,

```
LIST TO ARRAY ("Cities";aCities)
```

This code would be run in place of the assignment statements shown above.

If you need to save the user's choice into a field, you would use an assignment statement that runs after the record is accepted. A complete Case statement in the object method might look like this:

```
Case of
  :(Form event=On Load)
    LIST TO ARRAY ("Cities";aCities)
    If (Record number ([People])<0)  `new record
      aCities:=3  `display a default value
    Else   `existing record, display stored value
      aCities:=Find in array (aCities;City)
    End if
  : (Form event=On Clicked)  `User modified selection
      City:=aCities {aCities}  `field gets new value
  :(Form event=On Validate)
    City:=aCities {aCities}
  :(Form event=On Unload)
    CLEAR VARIABLE (aCities)
End Case
```

In the Events page of the Object Properties window, you would select each event that you test for in your Case statement.

Arrays always contain a finite number of items. The list of items is dynamic and can be changed by a method. Items in an array can be modified, sorted, and added to.

For information about creating and using an array, refer to the chapter on arrays in the *4th Dimension Language Reference* manual.

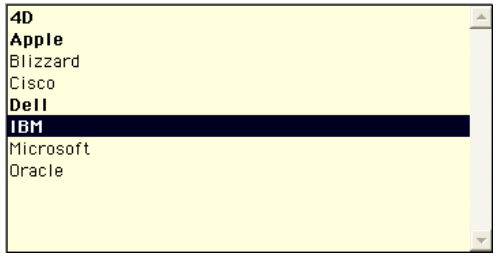
Goto Page Action

You can assign the Goto Page action to a pop-up menu/Drop-down list or a scrollable area. When that action is selected, 4th Dimension will automatically display the page of the form that corresponds to the number of the object that is selected in the drop-down list or scrollable area.

For example, if the user selects the third element of the list, 4th Dimension will display the third page of the current form (if it exists). If you want to manage the effect of the selection of one element, select **No action**.

Transparent scrolling areas

It is possible to make a scrolling area inserted in a form transparent. This allows, for example, you to create customized interfaces simulating multiple selections in this area type.



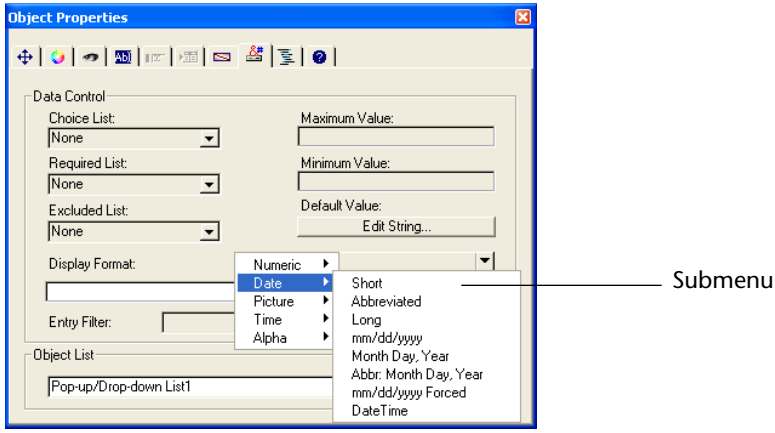
In this example, two scrolling areas are superimposed. The area located in the foreground has the “transparent” attribute and loads the list of values by default. The background area, empty by default, has a particular font style (bold). Depending on the line on which the user clicks, the element of one or the other area is displayed, thus simulating a selection/deselection.

Combo boxes

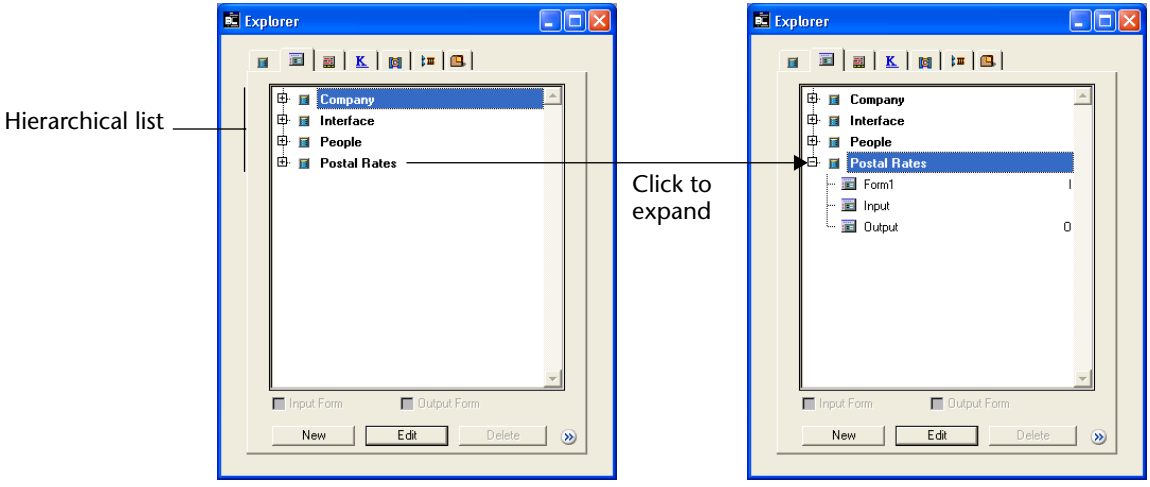
A Combo box is similar to a drop-down list, except that the object accepts text entered from the keyboard. You initialize a combo box in exactly the same way as a drop-down list. If the user enters text into the combo box, it fills the 0^{th} element of the array. In other respects, you treat a Combo box as an enterable area that uses its array as a set of default values. Use the On Data Change event to manage entries into the enterable area, as you would an enterable area object.

Hierarchical Pop-up Menus and Hierarchical Lists

A hierarchical pop-up menu has a submenu associated with each item in the menu. Here is an example of a hierarchical menu:



Similarly, a hierarchical list has a sublist associated with each item in the list. The Explorer is an example of a series of hierarchical lists:



In this example, each item in the list of tables has a sublist of forms. You can expand or collapse the hierarchical list by clicking on the plus or minus sign (Windows) or arrows (MacOS).

You can control whether an item in a hierarchical list can be modified by the user. If an item in a hierarchical list is modifiable, the user can hold down the **Ctrl** key (**Command** key under MacOS) to edit the text of the item.
If you populate a hierarchical list using a list created in the List editor, you control whether an item in a hierarchical list is modifiable using the Enabled/Editable check box in the List editor. For more information, see the paragraph "Enabled/Editable option," page 655.

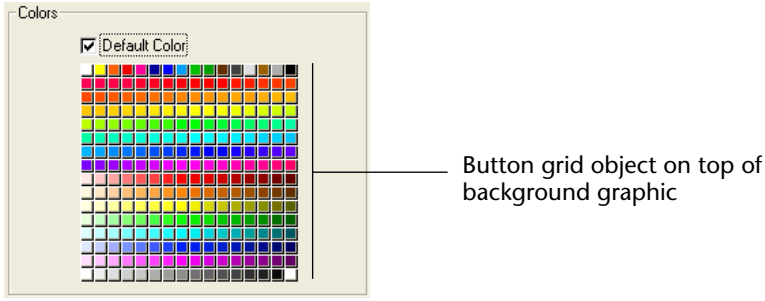
Note For reasons related to the internal management of hierarchical lists, it is not possible to place several hierarchical lists with the same name in the same form, even on different pages.

You manage hierarchical pop-up menus and hierarchical lists using the Hierarchical list commands in the language. For more information, see that section of the *4th Dimension Language Reference* manual.

Button Grids

A button grid is a transparent object that is placed on top of a graphic. The graphic should depict a row by column array. You can use a button grid object to determine where the user clicks on the graphic. Your object method would use the On Clicked event and take appropriate action depending on the location of the click.

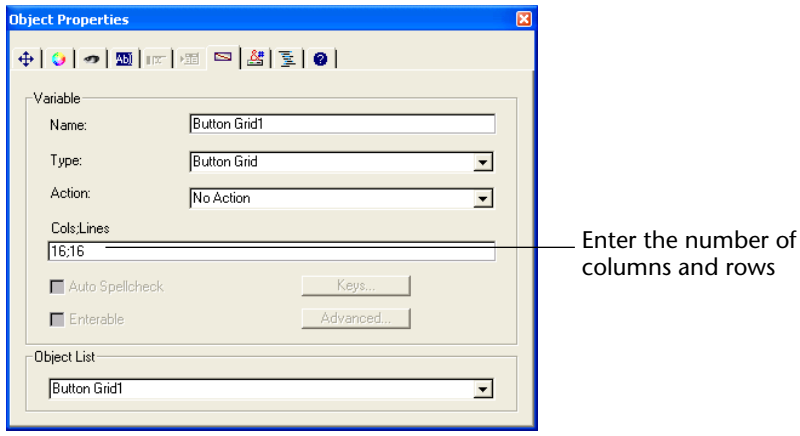
In 4th Dimension, a button grid is used as a color palette:



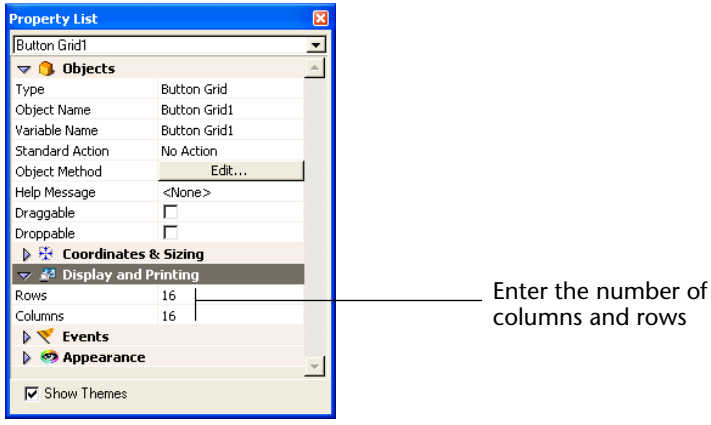
The buttons on the grid are numbered from top left to bottom right. In this example, the grid is 16 columns across by 16 rows down. The button in the top-left position returns 1 when clicked. If the red button at the far right of the second row is selected, the button grid returns 32.

To create the button grid, add a background graphic to the form and place a button grid on top of the graphic.

- In the Variable page of the Object Properties window, specify the number of rows and columns of the grid and the gap between elements of the grid.



- In the Property List, specify the number of rows and columns in the corresponding entry areas of the Display and Printing theme.



Goto Page Action

You can assign the Goto Page action to a button grid. When that action is selected, 4th Dimension will automatically display the page of the form that corresponds to the number of the button that is selected in the button grid.

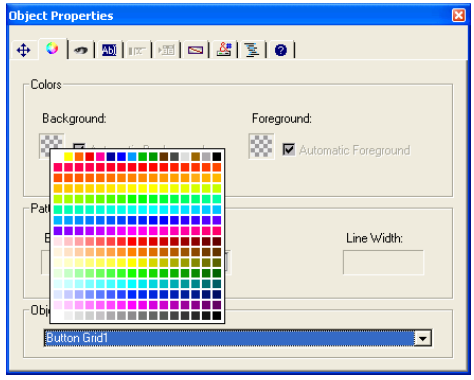
For example, if the user selects the tenth button of the grid, 4th Dimension will display the tenth page of the current form (if it exists).

If you want to manage yourself the effect of the selection of one button, select **No action**.

Picture Pop-up Menus

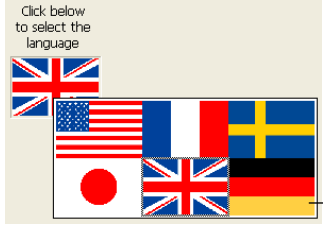
A Picture pop-up menu is a pop-up menu that displays a two-dimensional array of images. A picture pop-up menu can be used to replace a picture menu. The creation of the picture to use with a picture pop-up menu is similar to the creation of a picture for a picture button. The concept is the same as a button grid, except that the graphic is used as a pop-up menu instead of a form object.

The following illustration shows the color palette displayed as a Picture menu:



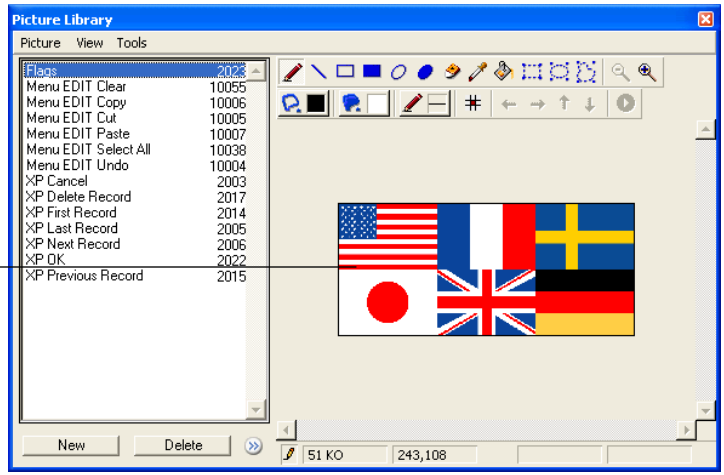
To create a Picture menu, you need to refer to an image. The following example uses the picture that was defined for picture buttons. In this case, it allows you to select the interface language by selecting it from a picture pop-up menu.

Each language is represented by the corresponding flag.



Picture pop-up menu

As with a picture button, a picture pop-up menu uses a picture that is organized in columns and rows. You can place that picture in the Picture Library, in a picture variable or in a PICT resource (MacOS).



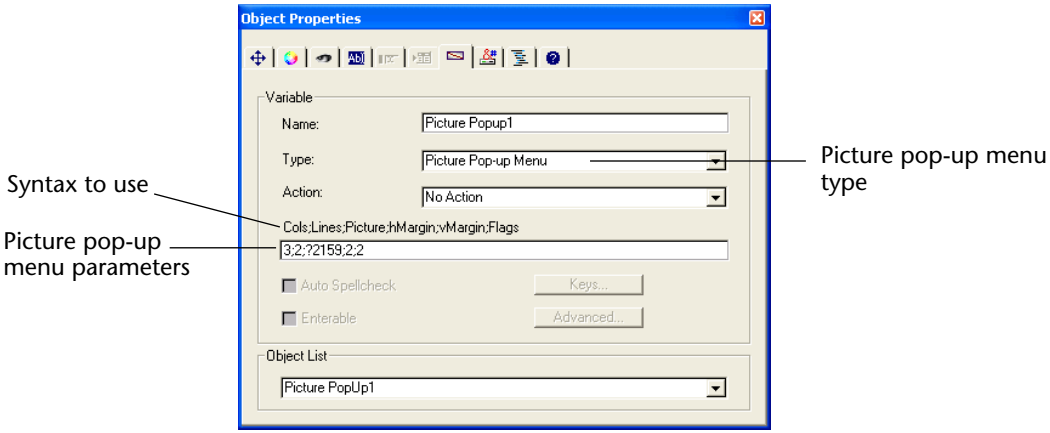
The Picture pop-up menu object displays the whole picture

Note The Picture Library includes features that allow you to organize a picture as a table of thumbnails. It also allows you to preview the effects of the current settings.
When a picture is defined as a table of thumbnails, you can create a picture pop-up menu by dragging the picture name into the form while pressing the Shift key.
For more information on the Picture Library, refer to Chapter 12, “Using the Picture Library” on page 659.

The method for setting the properties of a picture pop-up menu in the Form editor depends on whether you do it in the Object Properties window or in the Property List.

Using the Object Properties window

In the Object Properties window, select the Picture Pop-up Menu type in the Variable page. You should then set the parameters as follows:

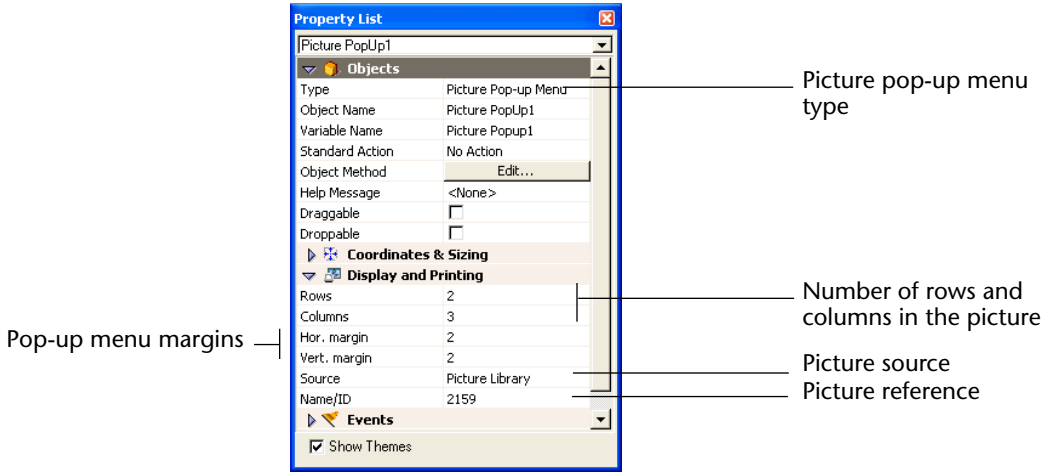


- The *Cols* and *Lines* parameters define the dimensions of the thumbnail table. Since this example uses three columns and two rows, the values are 3 and 2, respectively.
- The *Picture* parameter defines the picture that is used. It can be located in the Picture Library, in a variable, or in a PICT resource:
  - For a variable, enter the name of the picture variable,
  - For a PICT resource in your database, enter the PICT's reference ID as ":1234". Remember to include a colon, ":", before the picture's reference ID.
  - For a PICT in your Picture Library, enter the reference ID[1] of the picture preceded by a question mark, as shown in the example above.
- The parameters *vMargin* and *hMargin* create a gap between the border of the menu and the picture. Enter values in pixels.
- The parameter *Flags* allows you to specify whether the picture pop-up menu is transparent or not. The two values for *Flags* are:
  - 0 the pop-up menu box is not transparent, or
  - 64 the pop-up menu box is transparent.

1. The Reference ID is displayed in the Picture Library.

Using the Property List

In the Property List, select the **Picture Pop-up Menu** type in the Objects theme.



- In the Display and Printing theme, define the number of columns and rows of the thumbnail table.
- In the Display and Printing theme, select the source type from the Source drop-down list. You can choose between three source types: Picture Library, variable, and PICT resource. Once the picture source is set, enter the reference of the picture in the Name/ID entry area. The reference of the picture is its name if its source is a variable or a number if its source is a PICT resource of the Picture Library.
- The two entry areas Hor. Margin and Vert. Margin control the margins between the border of the menu and the picture.
- If you want the pop-up picture menu to be transparent, check that option in the Appearance theme.

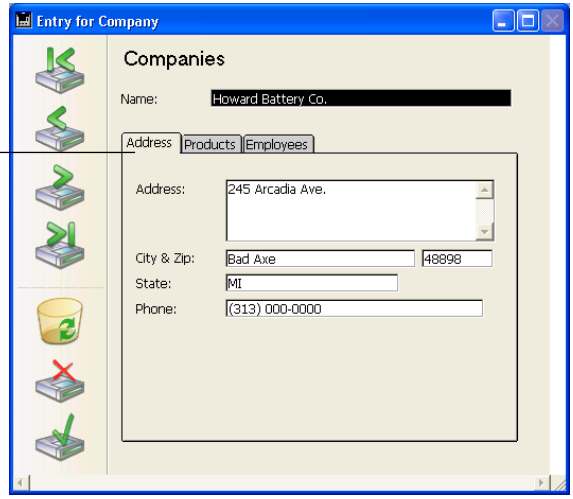
Operation of Pop-up Picture Menu

It is not possible to assign a standard action to a pop-up picture menu. Pop-up picture menus are managed using methods.

As for button grids, variables associated with pop-up picture menus are set to the value of the selected element in the pop-up picture menu. If no element is selected, the value is 0. Elements are numbered, row by row, from left to right starting with the top row.

Tab Controls

A tab control creates an object that lets the user choose among a set of virtual screens that are enclosed by the tab control object. Each screen is accessed by clicking its tab. The following multi-page form uses a tab control object:



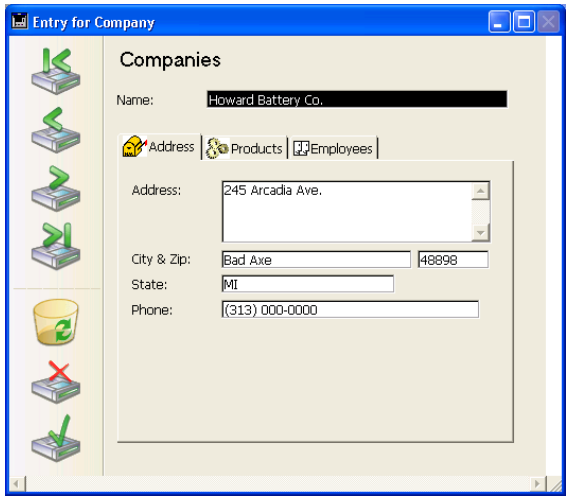
To navigate from screen to screen, the user simply clicks the desired tab.

The screens can represent pages in a multiple-page form or an object that changes when the user clicks a tab. If the tab control is used as a page navigation tool, then the GOTO PAGE command would be used when a user clicks a tab.

Another use of the tab control is to control the data that is displayed in a subform or grouped scrollable arrays. For example, a rolodex could be implemented using a tab control. The tabs would display the letters of the alphabet and the tab control's action would be to load the data corresponding to the letter that the user clicked.

Each tab can display labels or labels and a small icon. If you include icons, the icons appear to the left of each label.

Here is an example of a Tab control that uses icons:



When you create a tab control,4th Dimension manages the spacing and placement of the tabs. You only need to supply the labels in the form of an array or the icons and labels in the form of a hierarchical list.

If the tab control is wide enough to display all the tabs with both the labels and icons, it displays both.

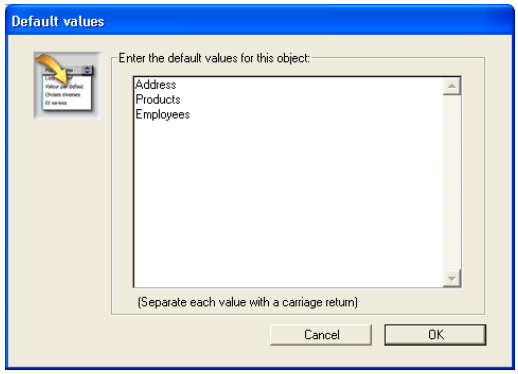
If the tab control is not wide enough to display both the labels and icons, 4th Dimension displays the icons only. If it can't fit all the icons, it places scroll arrows to the right of the last visible tab. The scroll arrows allow the user to scroll the icons to the left or right.

Under MacOS, once "Mac Theme" is set as the platform interface, the tab controls can be aligned, beyond just the standard position (top), to the left, to the right, or below.

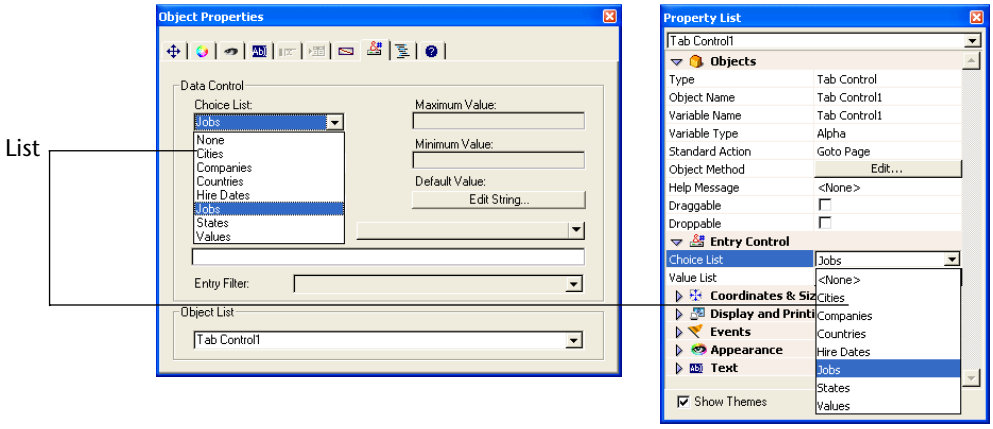
Adding Labels to a Tab Control

There are several ways to supply the labels for a tab control.

- In the Object Properties window, use the **Edit Strings** button on the Data Control page. In the Property List, use the **Edit** button for value list located in the entry control theme. Here is the Default Values dialog box that creates labels for the Products Tab control.



- Create a list using the List editor and assign the list to the tab control as a choice list, as shown below.



If you like, you can associate a small icon with each list element using the List editor. For more information about this, refer to the paragraph "Adding a Small Icon to an Item," page 650.

- You can create a Text array that contains the names of each page of the form. This code must be executed before the form is presented to the user. For example, you could place the code in the object method of the tab control and execute it when the On Load event occurs.

```
ARRAY TEXT (asPages;3)
asPage {1}:="Name"
asPage {2}:="Address"
asPage {3}:="Notes"
```

Note You can also store the names of the pages in a list and use the Load list command to load the values into the array.

You could also enter the names of the pages as default values using the Object Properties window. For more information, see "Default Lists of Values" on page 390.

- Use the GOTO PAGE command in the Tab control's method:

  GOTO PAGE (asPages)

  The command is executed when the On Clicked event occurs.

- You then clear the array on the On Unload event occurs.

  Here is an example object method:

```
Case of
  :(Form event=On Load)
     LIST TO ARRAY ("Tab Labels";asPages)
  :(Form event=On Clicked)
     GOTO PAGE (asPages)
  :(Form event=On Unload)
     Clear variable (asPages)
End Case
```

Goto Page Action

You can assign the Goto Page action to a tab control. When that action is selected, 4th Dimension will automatically display the page of the form that corresponds to the number of the tab control that is clicked.

For example, if the user clicks the third tab control, 4th Dimension will display the third page of the current form (if it exists). If you want to manage the tab control programmatically, select **No action**.

Modifying Direction (MacOS only)

You can define tab control direction in your forms. This function is available under two conditions:

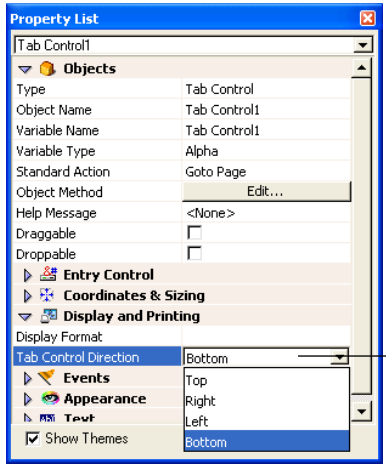
- The form is displayed in MacOS.
- **MacOS** is the "Platform" appearance property applied to the object.

Once these two conditions are met, you can choose to place the tab controls on **top** (standard), to the **left**, to the **right**, or on the **bottom**.

Tabs on top (standard layout)

Tabs on bottom

To define tab control alignment, you should display the Property list. The **Tab Control Direction** property is placed in the "Display and Printing" theme.



Tab Control Directions

When a form with customized tab controls is displayed under MacOS with an appearance other than "Mac Theme" or under Windows, the tab controls return to their standard location.

Plug-in Objects

A *plug-in object* is an area on the form that is completely controlled by a 4D plug-in written in C or Pascal.

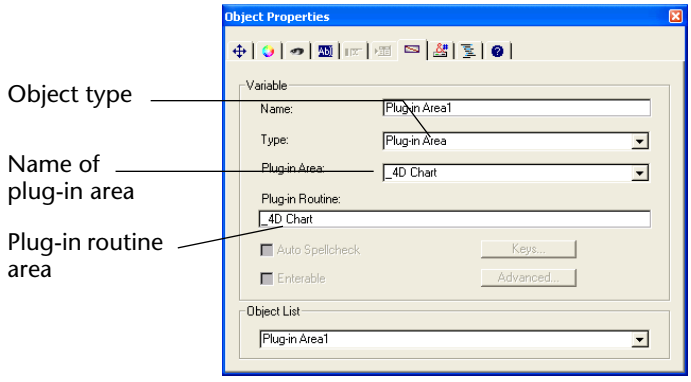
When opening a database, 4th Dimension creates an internal list of the plug-ins installed in your database. Once you have inserted a Plug-in Area in a form, 4th Dimension lists the available plug-ins in the Object Properties window. By default, 4D Chart, OLE tools (Windows only) and Report are available.

Note Some plug-ins cannot be used in forms or in external windows. When a plug-in cannot be used in a form, it does not appear in the plug-in list of the Object Properties window.

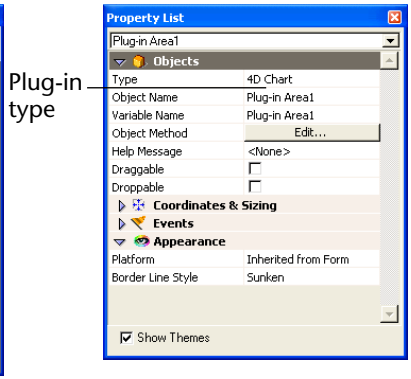
In the Object Properties window, you need to select the name of the plug-in from the Plug-in area list. When you select the plug-in from this drop-down list, 4th Dimension automatically inserts it in the Plug-in Routine enterable area.

In the Property List, the plug-in name is directly selected from the Type list:

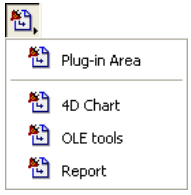
Object Properties window



Property List



Plug-in areas that are built into 4th Dimension can also be selected directly from the Tools palette:



Note When the object type is Plug-in Area, the Advanced button may be enabled. Advanced options may be provided by the author of the plug-in. If the plug-in provides additional options, you can click this button to set those options. Because the Advanced options dialog box is under the control of the author of the plug-in, information about the Advanced options is the responsibility of the distributor of the plug-in. If the plug-in has no Advanced options, the Advanced button is disabled.

Built-in plug-in areas

By default, 4th Dimension offers the following types of plug-ins:

- **4D Chart**: A 4D Chart area is used to display a chart in a form. There are numerous ways to calculate and display a chart area; nevertheless, they are all controlled by language commands. This subject is described in detail in the *Language Reference* manuals of 4th Dimension and 4D Chart.
- **OLE tools** (Windows only): Under Windows, an OLE area lets you open a window containing documents coming from other applications within your database. This subject is described in detail in the *OLE_Tools* manual.

Note It is also possible to create an OLE area directly using the **Insert an OLE Object** command in the **Objects** menu.

- **Report**: A Report area is used to insert a quick report into a form. The quick report area may display data in the form of a table, carry out summary calculations, etc. This area can be controlled using its own menu bar or using the language commands of 4th Dimension. For more information, refer to the *User Reference* and *Language Reference* manuals of 4th Dimension.

Installing Plug-ins

To install a plug-in in your 4D environment, you first need to quit 4th Dimension. Plug-ins are loaded when you launch 4th Dimension.

Under Windows, plug-ins are installed by copying the .4DX and .RSR plug-in files into a WIN4DX folder, located at the same level as the database structure file (.4DB), or next to the executable application, or yet again in the active 4D folder of your machine.

Under MacOS, plug-ins are installed by copying the plug-in file into a Mac4DX folder, located either at the same level as the database structure file (.4DB), or inside the software package, or in the active 4D folder of your machine.

Note The location of the active 4D folder will vary according to the operating system used. For more information, refer to the *4D Product Line Installation Guide*.

In addition, you will need to enter a serial or expansion number to be able to use certain plug-ins. For more information about the installation of plug-ins, please refer to the *4D Product Line Installation Guide*.

Using Plug-ins

The ability to incorporate plug-ins into forms gives you unlimited possibilities when creating custom applications. A plug-in can perform a simple task such as displaying a digital clock on a form, or a complex task such as providing full-featured word processing, spreadsheet, or graphics capabilities. Many of these capabilities are already available for the 4th Dimension environment in the form of 4D plug-ins. For example:

- 4D Write, a word processing application,
- 4D View, an advanced spreadsheet and list management application,
- 4D Draw, an object-oriented drawing application.

For more information, refer to the documentation that comes with the 4D Productivity Plug-ins.

If you are interested in designing your own plug-ins, you can receive extensive information about writing and implementing plug-ins. 4D provides the following information sources:

- **Platform Independent 4D Plug-ins API** The 4D Plug-ins API teaches you how to write Plug-ins for 4th Dimension on both Windows and Macintosh.
- **Developer Tools** These developer mailings provide information and examples of plug-ins that can be used and modified for the developer's own needs.

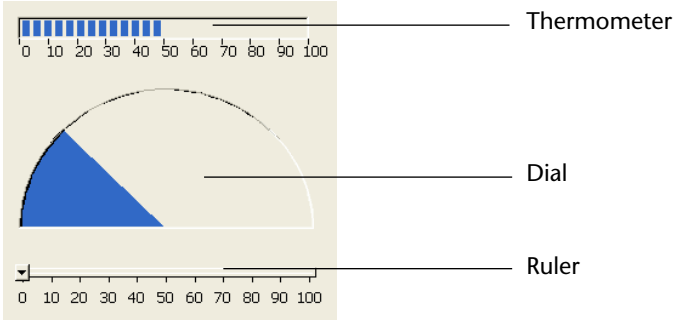
For more information, contact 4D, Inc. (http://www.4d.com)

Indicators

Thermometers, rulers, and dials are objects that display a value graphically. The three objects work in the same way; they differ only in their appearance. We refer to these three objects as *indicators*.

You can use indicators either to display or set values. For example, if a thermometer is given a value by a method, it displays the value. If the user drags the indicator point, the value changes.

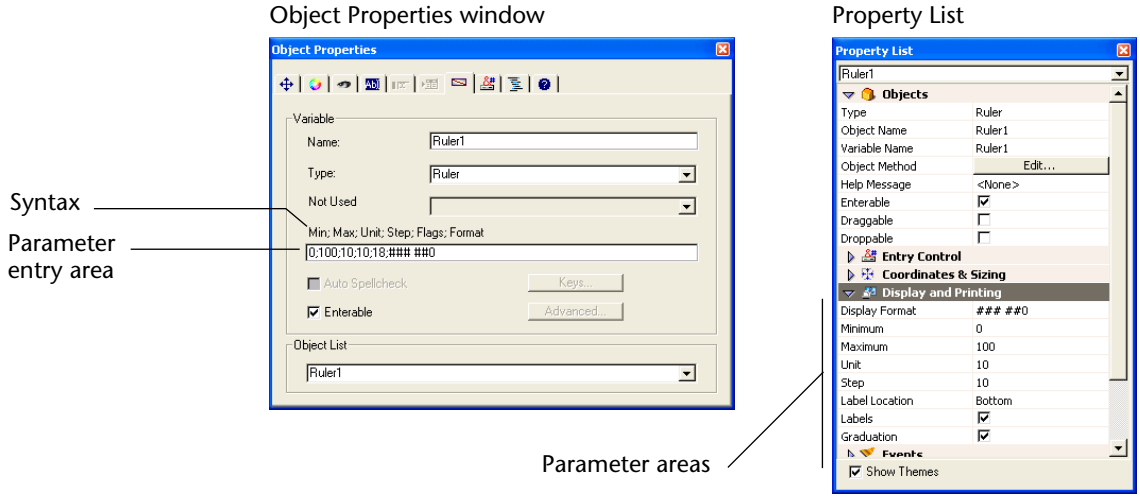
The value can be used in another object such as a field or an enterable or non-enterable object.



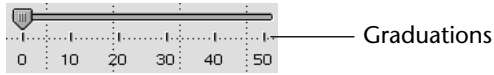
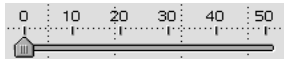
In addition to the standard positioning and appearance settings, you can set some other specific properties for indicators: minimum value, maximum value, units for the tick marks, the minimum steps permitted by the indicator as well as display options. You can also define the display format of an indicator's label (for more information on display formats, refer to "Display Formats" on page 396).

In the Object Properties window, those properties are defined using a sequence of parameters. In the Property List, those properties as well as

the display options are specified in the “Entry Control” and “Display and Printing” themes.



Various display options can be defined only for rulers and thermometers (except for the last option, that is also accessible to dials). In the Object Properties window, these options are set using the *Flags*[1] parameter.

Value for Flag	Property List Options	Description
0	No option selected	Does not display the units.
2	Labels + Label Location= Bottom	Display the units on the right or below the indicator. 
3	Labels + Label Location= Top	Display the units on the left or above the indicator. 
16	Graduation	Display graduations adjacent to the units, as shown in both illustrations above. Used only if 2 or 3 is also selected.
32	Execute Object Method	On Data Change is executed while the user is adjusting the indicator. If this value is not used, On data change occurs only after the user is finished adjusting the indicator.

1. The *Flags* parameter is a sum of different options. If, for example, the value for Flags is 51 (32+16+3), the indicator will display the indicator’s units and the object method be dynamically executed.

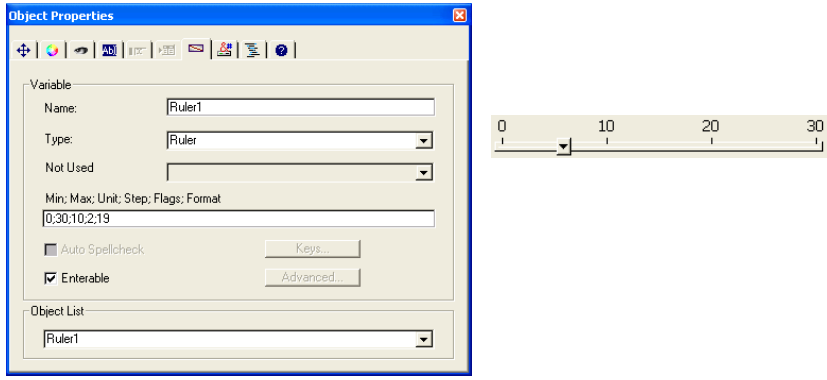
Note Option 2 and option 3 cannot be used at the same time.

For example, if you enter:

0;30;10;2;19

the object sets the minimum to 0, the maximum to 30, places tick marks every 10 units on the indicator, increments the display when the value changes 2 units, and displays the graduation with the labels on top (options 16+3).

The figure below shows these values being set for a thermometer.



The variable associated with the indicator controls the display. You place values into, or use values from the indicator using methods. For example, a method for a field or enterable object could be used to control a thermometer.

The method:

vTherm:=[Employees]Salary

assigns the value of the Salary field to the vTherm variable. This method would be attached to the Salary field.

Conversely, you could use the indicator to control the value in a field. The user drags the indicator to set the value. The method:

[Employees]Salary:=vTherm

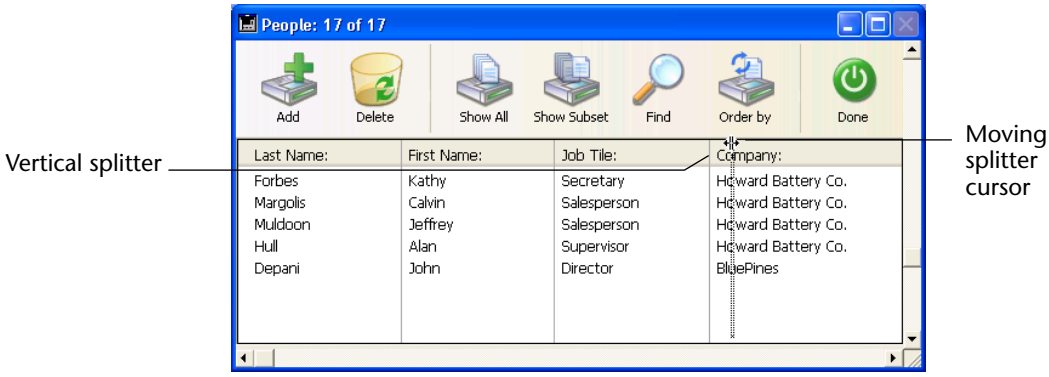
assigns the value of the thermometer to the salary field. As the user drags the indicator, the value in the Salary field changes.

Splitters

A splitter divides a form into two areas, allowing the user to enlarge and reduce the areas by moving the splitter one way or the other. A splitter can be either horizontal or vertical.

The splitter takes into account each object's resizing properties, which means that you can completely customize your database's interface.

The splitter is generally used in output forms (in the Custom Menus environment) so that columns can be resized:



Here are some of the splitter's general characteristics:

- You can place as many splitters as you want in a form and use a mixture of horizontal and vertical splitters in the same form.
- A splitter can cut (overlap) an object. This object will be resized when the splitter is moved.
- Splitters cannot be used in output forms in the User environment or in subforms.
- If you resize a form using a splitter, the new dimensions of the form are saved only while the form is being displayed. Once a form is closed, the initial dimensions are restored.
- Once it is inserted, the splitter appears as a line. You can modify its border style to obtain a thinner line or change its color. Place the splitter on your form according to your needs.

Interaction with the properties of neighboring objects

In a form, splitters interact with the objects that are around it according to these objects' resizing options:

Resizing options for the object(s)	Object(s) above the horizontal splitter or to the left of the vertical splitter[1]	Object(s) below the horizontal splitter or to the right of the vertical splitter[2]
None	Remain as is	Are moved with the splitter; their position relative to the splitter is not modified
Resize	Keeps their original position, but they are resized according to the splitter's new position	
Move	Are moved with the splitter	

1. You cannot drag the splitter past the right (horizontal) or bottom (vertical) side of an object located in this position.
2. The buffer, when moving a horizontal splitter toward the bottom or a vertical splitter to the right is either the window's border or another splitter. This buffer is calculated in such a way that the moved objects remain entirely visible in the form or do not pass under/next to another splitter.

Note An object completely contained in the rectangle that defines the splitter is moved at the same time as the splitter.

Managing splitters programmatically

You can associate an object method with a splitter and it will be executed when the user releases the splitter.

A variable of type Longint is associated with each splitter. This variable can be used in your object and/or form methods. Its value indicates the splitter's current position, in pixels, relative to its initial position.

- if the value is negative: the splitter was moved toward the top or toward the left,
- if the value is positive: the splitter was moved toward the bottom or toward the right,
- if the value is 0: the splitter was moved to its original position.

You can also move the splitter programmatically: you just have to set the value of the associated variable. For example, if a vertical splitter is associated with a variable named *split1*, and if you execute the following statement: split1:=-10, the splitter will be moved from 10 pixels to the left — as if the user did it manually.

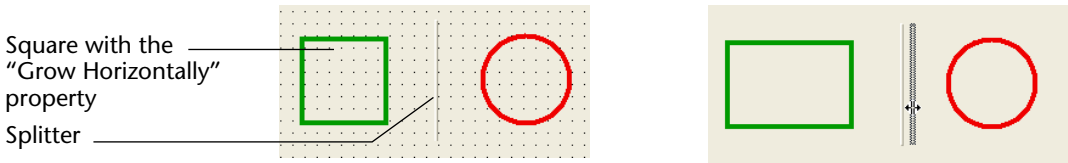
The move is actually performed at the end of the execution of the form or object method containing the statement.

The Automatic Splitter action allows you to create custom splitters in your forms. You can assign this action to an object of type **invisible button**. When an invisible button is assigned this standard action, it acts exactly as a splitter. If, for example, you paste a picture on the invisible button, you can create a custom interface for your splitters. For more information about this type of button, refer to "Buttons" on page 421.

Examples

This section gives some practical examples of the use of splitters.

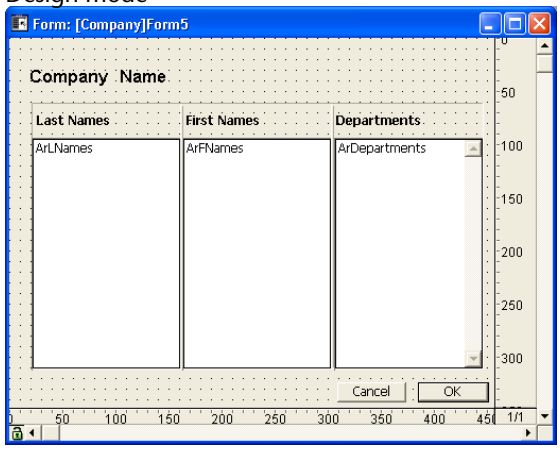
- *Example 1*: An input form contains a square, a vertical splitter and a circle. The "Grow Horizontally" property is applied to the rectangle. When the splitter is moved to the right or to the left, the rectangle becomes bigger or smaller, the circle is moved and its size is not modified. If the window is resized, the objects do not change position or size.



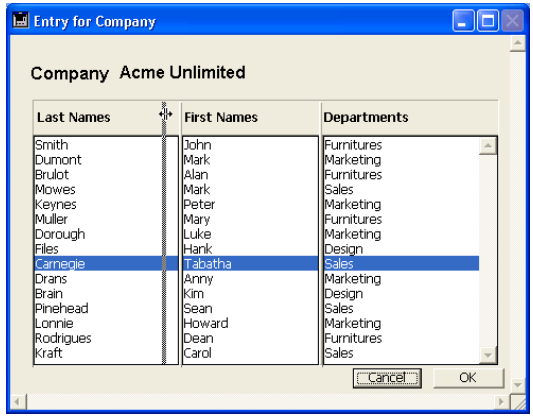
- *Example 2*: An input form containing three grouped arrays. The "Horizontal sizing" property is applied to all three arrays as well as their titles and a vertical splitter is placed between each column.

  In this way, you can modify each columns' relative size:

Design mode



User mode



Duplicating on a Matrix

Sometimes you may want to place several similar active objects in a form at the same time, numbering them sequentially so that their names are unique. For example, you may want to create a series of buttons that perform database operations. Duplicating on a matrix has the additional advantage of quickly and easily aligning multiple objects.

You can either duplicate an active object on a matrix manually or use the Duplicate Many dialog box, that allows you to quickly populate the matrix.

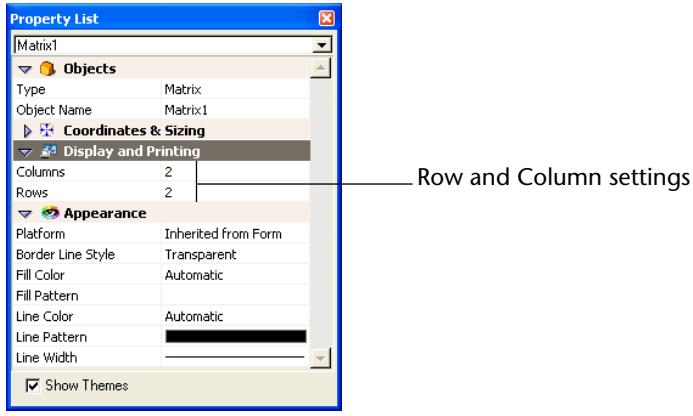
▶ To duplicate an object on a matrix:

1 **Select the Matrix tool in the Tools palette (variation of the Rectangle group) and create a matrix on the form.**

You can either drag the Matrix tool onto the form or draw the matrix itself. Make sure that each cell in the matrix is large enough to contain the object you want to duplicate.

2 **Display the Property List and select the matrix you just created.**

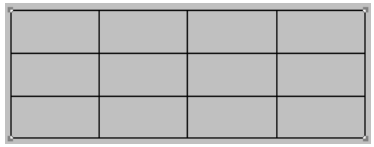
3 **In the Display and Printing theme, set the number of columns and rows of the Matrix.**



4 **If necessary, set the appearance of the matrix using the options in the Appearance theme.**

You can set the platform interface, the border style, the line thickness, the color, and the fill pattern.

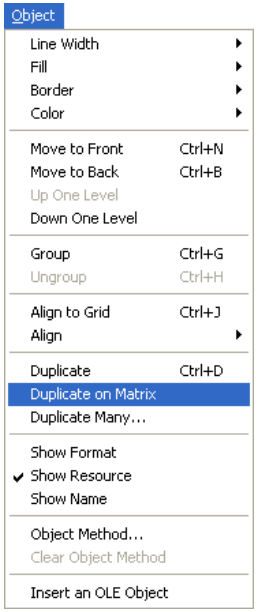
5 Make sure the cells in the matrix are larger enough to contain object that you want to duplicate.



For more information about resizing objects, refer to Chapter 5.

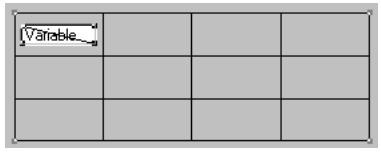
6 Create a new active object and place it in the upper left cell of the matrix.

Make sure that the object type matches the type and size of the object you want to duplicate.



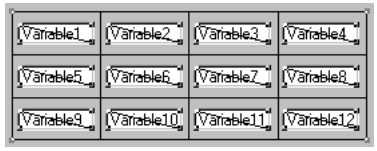
7 Name the object and make sure its name does not end with a number.

8 Select both the object and the matrix.



9 Choose Duplicate on Matrix from the Object menu.

4th Dimension copies the active object to each cell in the matrix, giving each active object a unique number.



The objects are copied along with their size and style properties as well as their associated method (where applicable).
4th Dimension numbers the active objects from top to bottom in each column. These numbers are added to the object name for each object, thus creating a unique object at every matrix cell.

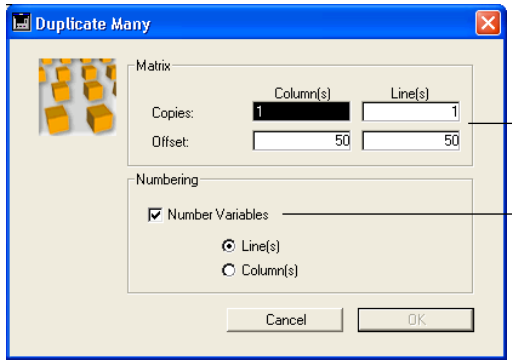
Note To number the series of active objects from left to right in each row, hold down the **Alt** key (Windows) or **Option** key (MacOS) when you choose **Duplicate on Matrix**.

You can refer to these objects in methods using the names they have been given. You can delete the matrix or leave it in the form.

► To duplicate one or several objects using the Duplicate Many dialog box:

1 **Select the objects you want to duplicate.**

2 **Select Duplicate Many from the Object menu.**



3 **In the Matrix area, enter the number of rows and columns as well as the offset between them.**

For more information about this point, refer to "Duplicating Objects" on page 326.

4 **Select the "Number Variables" option.**

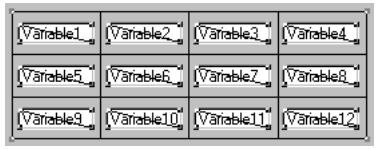
This option is enabled only if you selected a variable.

5 **Select the order of the numbering.**

If you select the **Line(s)** option, 4th Dimension will number active objects from top to bottom (column by column) and left to right. f you select the **Column(s)** option, 4th Dimension will number active objects from left to right (line by line) and top to bottom.

6 **Click the OK button.**

The objects are copied and numbered according to your settings:



Using Object Methods with Fields and Objects

You can attach a method to any active object in a form. Methods that are attached to individual objects on a form are called *object methods*.

The following are some of the more common uses of object methods:

- Enforce data entry constraints,
- Initialize and manage interface objects such as tab controls, pop-up menus, drop-down lists, combo boxes, hierarchical lists, and pop-up menus.
- Specify the action that takes place when an object is clicked or double-clicked,
- Manage drag and drop operations.

The previous section on types of active objects gives several simple examples of how methods are used to manage various kinds of objects. Here are some simple examples that perform operations on data:

The following method calculates a total based on data in two other fields:

```
Line Total:= Price * Quantity
```

Here is a method to make all characters in a Name field uppercase:

```
Name:= Uppercase (Name)
```

Note By default, built-in functions are displayed in boldface and user-written functions are displayed in italics. For a list of built-in functions, see the *4th Dimension Language Reference* manual.

The following method concatenates values from a First Name field and a Last Name field and assigns the results to a variable named vName:

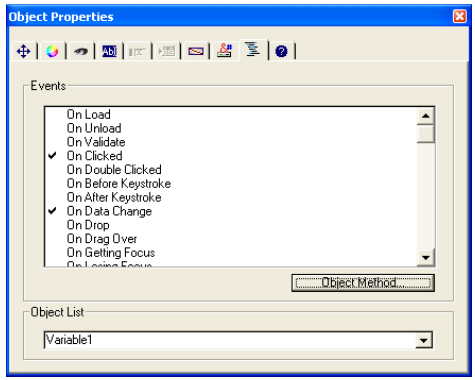
```
vName:= First Name + " " + Last Name
```

Because each object method is attached to its object, you create object methods from within the Form editor. For information on how to use the Method editor, see the section "Using the Method Editor" on page 548.

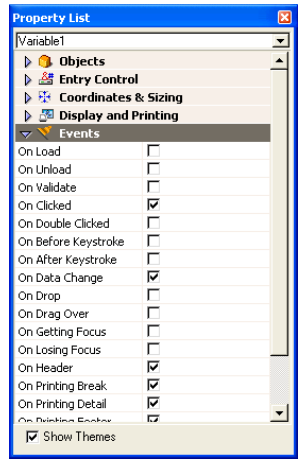
Object Events

Object methods run when certain events occur. For example, the action associated with a tab control makes sense only when a user clicks a tab. In a scrollable area, you may want the method to execute only when the user double-clicks an item. You can specify which events will be executed for a particular object in the Events page of the Object Properties window or in the Property List.

Object Properties window



Property List



The Events list displays all possible events that can lead to the execution of the object method. Here are the events that are relevant for forms that are displayed on-screen:

- **On Load** 4th Dimension is about to display the form on-screen or print the form.
- **On Unload** The form is about to be closed and released.
- **On Validate** After the user clicks OK or a navigation button.
- **On Clicked** The user clicks the object.
- **On Double Clicked** The user double-clicks the object.
- **On Before Keystroke** The user just entered one character in the object that has the focus. The Get Edited Text command returns the object's contents without that character.
- **On After Keystroke** The user just entered one character in the object that has the focus. The Get Edited Text command returns the object's contents with that character.

- **On Data Change** When the user changes the value of an object.
- **On Drop** When the user drops an object on a dropable object.
- **On Drag Over** When a dragged object can be dropped on the object (i.e,. the dragged object is in position to be dropped on the object).
- **On Getting Focus** When a form object gets focus (i.e., the user presses Tab to select the object or clicks the object to select it).
- **On Losing Focus** When a form object loses focus (i.e., the user presses Tab to select the next object in the entry order or clicks another object to select it).
- **On Timer** The number of ticks, set using the SET TIMER command was reached.
- **On Header** The form header is about to be printed or displayed.
- **On Printing Break** A form break is about to be printed.
- **On Printing Detail** A form detail area is about to be printed.
- **On Printing Footer** A form footer area is about to be printed.
- **On Display Detail** A record is about to be displayed in an output form.
- **On Plug-in Area** A plug-in area triggers the execution of its object method.
- **On Outside Call** When the form receives a call from CALL PROCESS.

Note 4th Dimension also uses form events that are not available from the Object Properties window or the Property List. For a complete description of events in 4th Dimension, refer to the *4th Dimension Language Reference* manual.

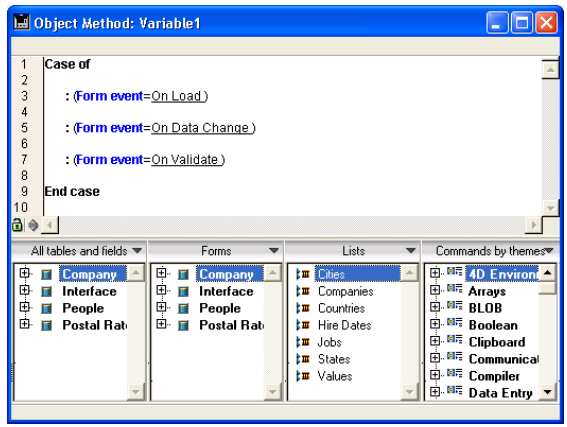
You activate an event by clicking the desired event. A check mark appears to the left of the events you select.

To select or deselect all the events at once, Press the **Ctrl** key (Windows) or the **Command** key (MacOS) while clicking any event.

If you need to execute different code segments for several different events, use a Case statement in your method and test for each event you checked in the Events page. To test for an event, you use the Form event function and the Form Event constants in the Constants page of the Explorer.

You can add a constant to your code by opening the Explorer to the Constants page and dragging the desired constant to the desired point in the code. By default, all constants are underlined in the Method editor.

An example shell for an object method might look like this:



For more information on object methods, see "Object Methods" on page 518 and the section on Form Events in the *4th Dimension Language Reference* manual.

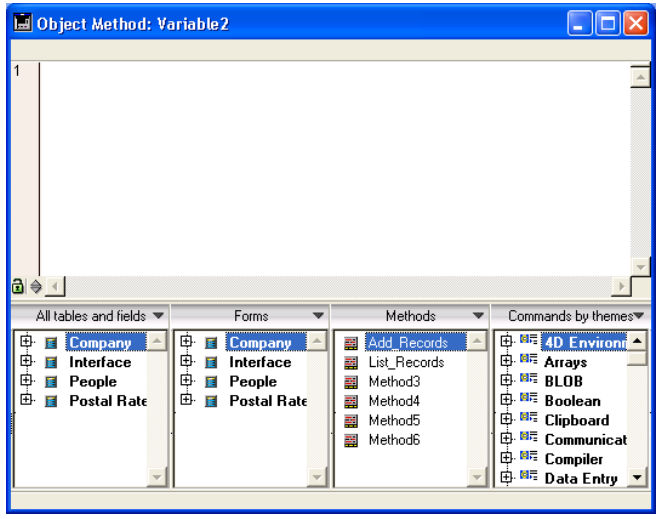
► To add an object method:

1 **In the Form editor, select the object to which you want to assign a method.**

2 **Click the Object Method button in either the Events page or the Field page of the Object Properties window.**
OR
Click the Edit button located next to the Object Method line in the Property List.
OR
Choose Object Method from the Objects menu.
OR
Under Windows, click the object using the right mouse button and choose Object Method from the contextual menu. Under MacOS, click the object while pressing the Control key and choose Object Method from the contextual menu.
OR
Hold down the Alt key (Windows) or Option key (MacOS) and click the field or object.

If you are creating a new method and if you have not set a default editor type in the Preferences dialog box, 4th Dimension displays the Method Type dialog box where you can select the preferred Method editor.

4th Dimension displays a blank Method editor window. The name of the new window includes the word “Method” and the name of the object or field.



Note If an object or field is grouped with another object, it must be ungrouped before you can open its method.

3 Write the method.

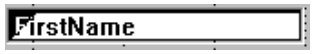
You can use several techniques to enter text into the Method editor. You can:

- Type text into the editor,
- Double-click on field or table names, methods, commands or macros in the scrollable areas below the text area,
- Drag table names, field names, form names, constants, built-in commands, plug-in commands, or project methods into the editor from the Explorer.

For more information on how to use the Method editor, see the section “Using the Method Editor” on page 548.

4 Close the Method window (optional).

The method is now associated with the field or active object. When an object has a method, a triangle appears in the top left corner of the object.



You can view or modify a method at any time.

▶ To open a method for viewing or modification:

1 **In the Form editor, select the object whose method you want to open.**

2 **Click the Edit button located next to the Object Method line in the Property List.**
OR
Hold down the Alt key (Windows) or the Option key (MacOS) and click the object to which the method is attached.
OR
Choose Object Method from the Objects menu.
OR
Under Windows, click the object using the right mouse button and choose Object Method from the contextual menu. Under MacOS, click the object while pressing the Control key and choose Object Method from the contextual menu.
OR
Click the Object Method button in the Field or Events page of the Object Properties window.
The Method editor appears with your method, ready for you to make any changes.

Deleting an Object Method

If you create an object method and then find that you do not need it, you can use the **Clear Object Method** item in the **Object** menu to remove the method. To do so, select the object to which the method is attached and then choose **Clear Object Method** from the **Object** menu.

You can also click on the Clear Object Method button in the Tools palette:  .

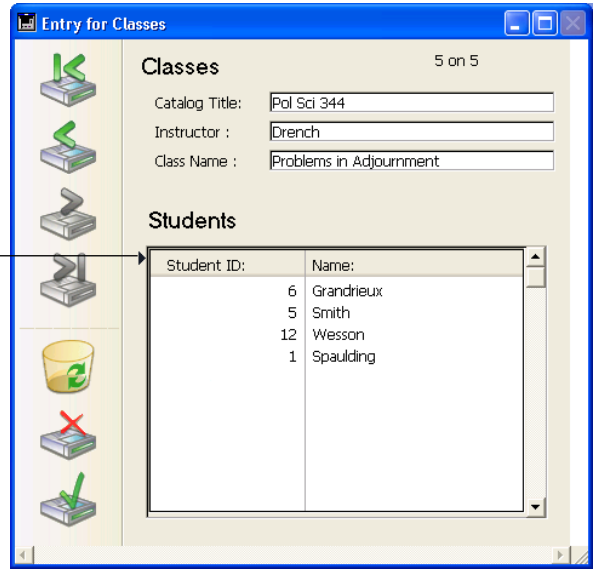
Adding a Subform to the Form

A subform is a List form from another table or subtable in the Master table that is displayed in a Detail form. A subform lets you enter, view, and modify data in other tables. You usually use subforms in databases in which you have established One-to-Many relations. A subform on a form in a related One table lets you view, enter, and modify data in a related Many table. You can have several subforms on the same form and they can belong to different tables or subtables.

For example, a Contacts manager database might use a subform to display all the telephone numbers for a particular contact. Although the telephone numbers appear on the Contacts screen, the information is actually stored in a related table.

Using a One-to-Many relation, this database design makes it easy to store an unlimited number of telephone numbers per contact. With automatic relations, you can support data entry directly into the related Many table without programming.

In the figure below, a subform in a Detail form has focus. A small flashing triangle points to the subform title bar and the Add to Subform button is enabled[1].



Flashing triangle indicates that the subform has focus

1. If you have more than one subform on a Detail form, the subform that has focus has a small flashing triangle at its top left corner.

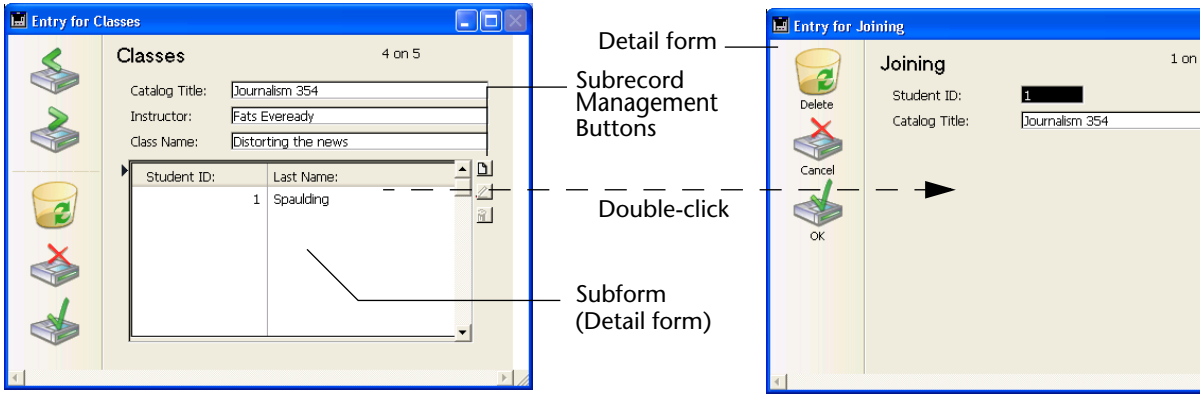
To add a record to the related Many table, the user clicks the **Add to Subform** button or presses **Ctrl+/** (**Command–Tab** under MacOS[1]).

You can create a subform with the Form Wizard when you create a new form or you can add one or more subforms to an existing form using the Form editor. You must have first created the List form you want to use.

Adding a subform using the Form Wizard is described in the section "Adding a Subform to the Form" on page 255.

Entering Data in a Subform

Subforms can be used for data entry in two ways: the user can enter data directly in the subform or enter it in an input form. The figure below shows both a subform and an input form associated with it.

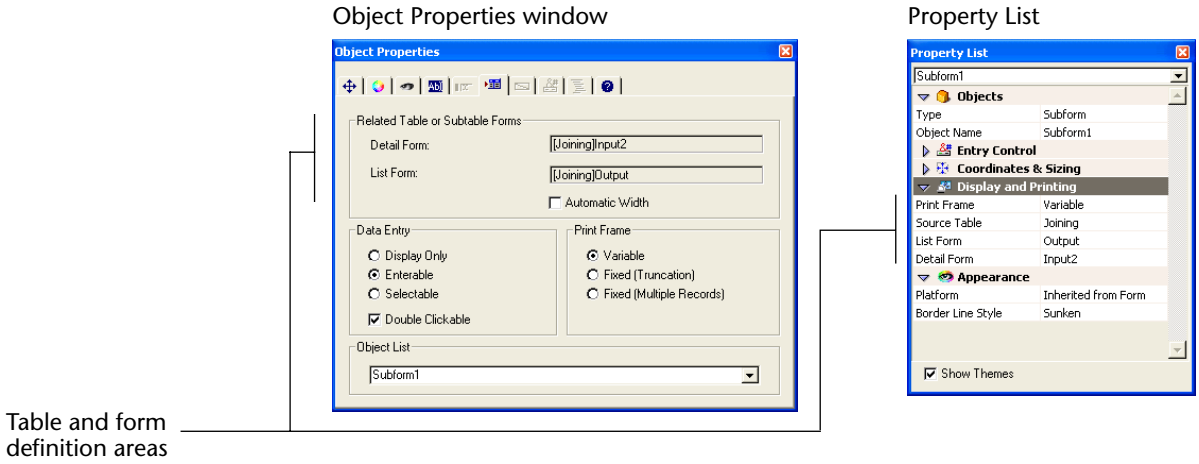


The form used as the subform is referred to as the *List form*. The input form is referred to as the *Detail form*.

You can allow the user to enter data through the List form and you can allow the user to double-click a row in the List form to display the Detail form.

1. These keyboard hotkeys can be modified using 4D Customizer Plus

The following illustration shows the Subform page of the Object Properties window and the Property List:.



The Related Table or Subtable area shows the List form and Detail form for the subtable area. You specify these forms by dragging them from the Forms page of the Explorer to the subform area on the form. In the Property List, you can specify those forms by dragging them from the Explorer Window.

Data Entry Options for Subforms

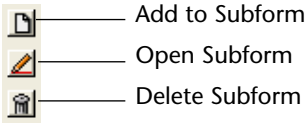
The Data Entry area in the Subform page and the Display and Printing theme on the Property List includes the following controls:

- **Enterable** Allows data entry in the List form. If **Enterable** is not checked, the List form is used for display only. When the **Double-clickable** option is selected, the **Enterable** option also allows the user to edit the subrecord in the input form.
- **Display Only** Allows the user to view the records in the List form but not perform data entry. If this option is selected, the user will not be able to display the Detail form, even if the **Double-clickable** option is selected.
- **Selectable** During data entry, clicking on a selectable subform highlights an entire record — similar to highlighting a record in an output form. However, unlike its behavior in an output form, the selected record becomes the current record for its table (or subtable).
- **Double Clickable** Allows data entry in the full page form. If **Double Clickable** is not checked, the user cannot use the Detail form.

You normally use an output form as the List form and an input form as the Detail form. If you do not specify a Detail form, 4th Dimension automatically uses the default input form for that table.

You can add custom buttons to control data entry for records in a subform. These button actions are **Open Subform**, **Delete Subform**, and **Add To Subform**.

Any type of button—Button, Highlight, or Invisible—can be used. For more information about adding these buttons, refer to "Buttons" on page 421.



Displaying Data in a Subform

A subform typically displays related Many records using an automatic One to Many relation.

4th Dimension places a scroll bar on the right of the subform so that you can scroll through the list of records displayed. The scroll bar extends all the way up the right side, even alongside the column headings in the subform.

You can adjust the subform as necessary to display the records. The wider you make the area, the more columns can be displayed. The taller you make the area, the more records can be displayed. A subform area automatically displays a scroll bar so that the user can scroll through the records or subrecords.

Creating a Subform

You can create a subform in two ways:

- In the Form Wizard, using the Subform page in Advanced options,
- In the Form editor, using the form and the Forms page of the Explorer.

For information on the Subform page of the Form Wizard, see the section "Adding a Subform to the Form" on page 255.

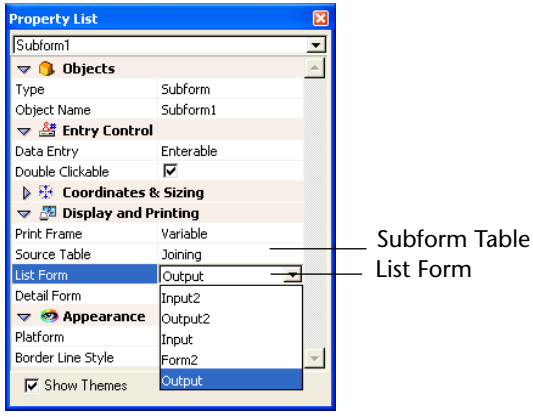
► To create a subform in the Form editor:

1 **Create a subform object using the subform tool**  **.**

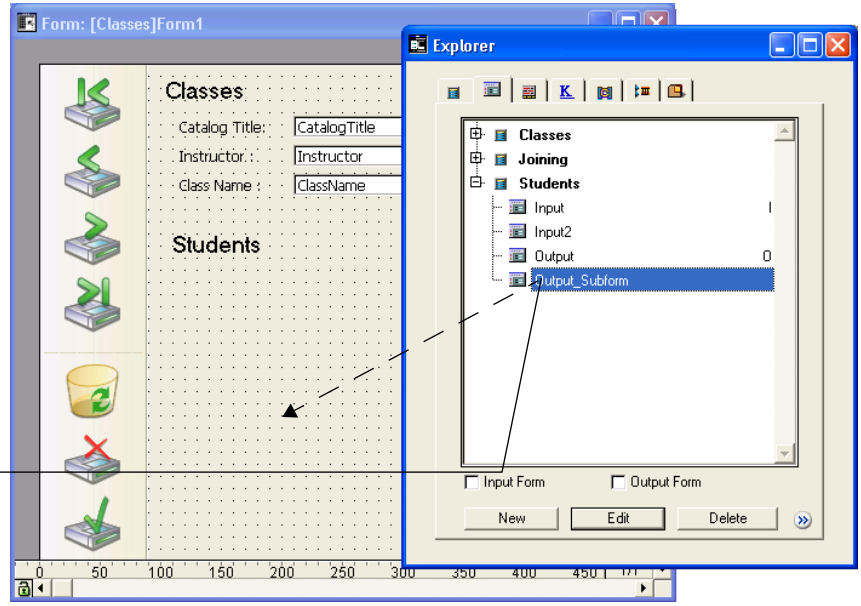
You can either drag the tool from the Tools palette or select the tool and draw the subform area in the form (see "Creating Objects" on page 312).

2 In the Display and Printing theme of the Property List, select the source table for the subform.
OR
Open the Forms page of the Explorer and expand the table that contains the records you want to display.

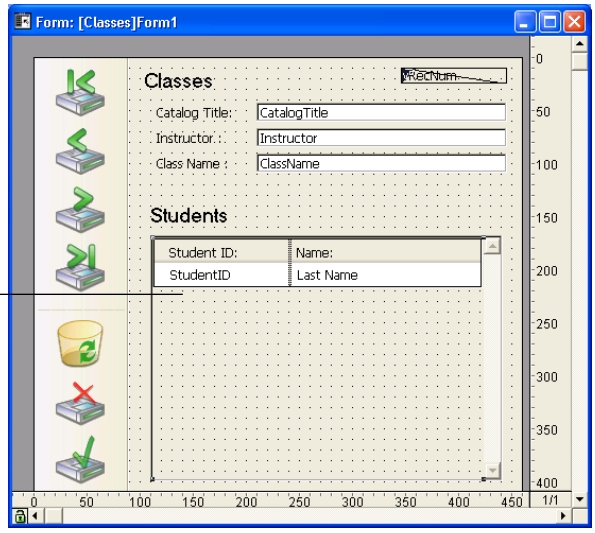
3 In the Property List, select the List form that you want to use.



OR
Drag the List form you want to use as the subform from the Forms page of the Explorer to the input form.



4th Dimension creates a subform area on the form and displays the List form in the subform area.

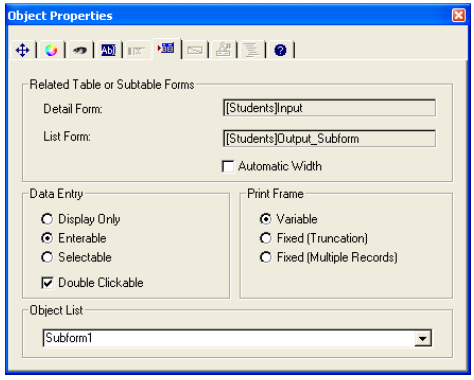


Note You can create a subform directly by dragging the subform from the Explorer. In this case the width of the subform will be defined by 4th Dimension automatically (for more information refer to "Setting the Width of a Subform" on page 474).

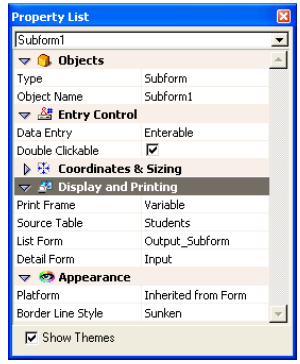
4 **If you want the subform area to be double-clickable, select the Detail form you want to use in the Property List.**
OR
Hold down the Shift key and drag the name of the Detail form you want to use from the Forms page of the Explorer to the subform area on the form.

The names of both forms appear in the Related Tables area on the Subform page of the Object Properties window.

Object Properties window



Property List



Dragging the Detail form name does not change the appearance of the subform area on the form itself.

If you drag the wrong List or Detail form to the subform area, you can replace it by dragging the correct form.

Setting the Type of Data Entry for the Subform

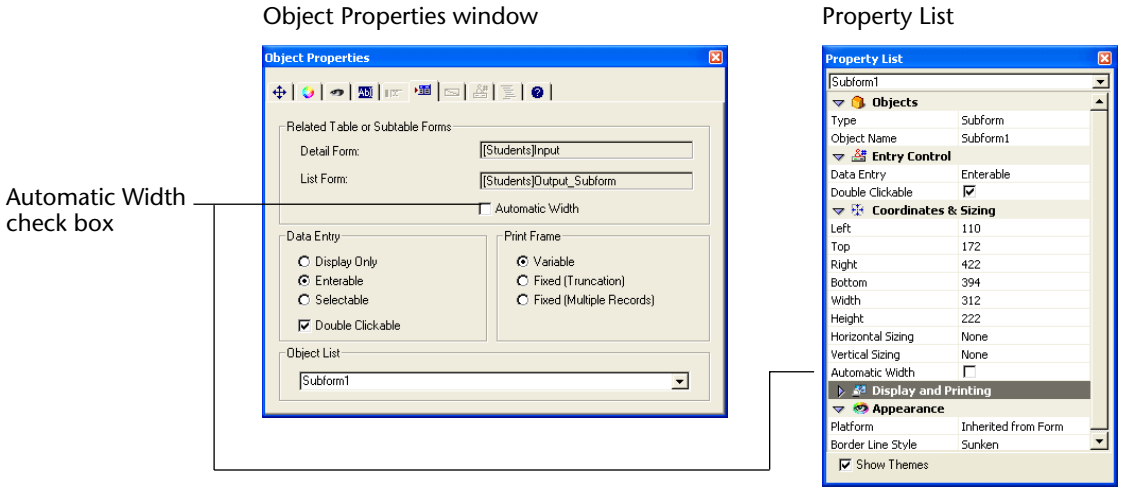
You can use the **Enterable**, **Display Only**, **Selectable** and **Double-clickable** radio buttons to specify the kind of data entry you want to have available for the subform.

For more information on data entry in subforms, refer to "Entering Data in a Subform" on page 469.

Setting the Width of a Subform

You can resize a subform area as you can any other object or you can let 4th Dimension automatically sizes the subform area so that all the fields in the List form are displayed.

To select 4th Dimension's automatic sizing feature, you can use the Automatic Width property to prevent designers from changing the width of a subform area.



Note When you create a subform area by dragging it directly from the Explorer window, 4th Dimensionautomatically sets the width of the subform.

If you click the Automatic Width check box, you only change the height of the subform area when you drag a resizing handle, leaving the width correct. Changing the height allows you to display more or fewer records.

Changing other Subform Properties

After you have specified the List and Detail forms for the subform and set the data entry and printing properties, you can set other properties of the subform area. The other pages of the Object Properties window work as for any other active object type. For example, you can set the platform interface, appearance, and sizing and resizing options. Use the other pages of the Object Properties window to set any other properties you like. For more information, see "Setting Object Properties" on page 412.

For more information on printing options for forms, refer to "Printing Subforms, Pictures, and Text Fields" on page 509.

Modifying a Subform source

You can modify the "source" List or Detail form of a subform area at any time:

► To modify the source form of a subform:

1 **Double-click the subform area to display the source List form.**
OR
Shift+double-click the subform area to display the source Detail form.
4th Dimension displays the desired source form in a new window of the Form editor.

2 **Make any necessary changes and save the form.**
For example, you may need to resize the columns of the List form or insert another field in the Detail form.
Refer to chapter 5, "Form Editor Basics," page 267 for information about form editing.

7 Output Displays and Reports

Output forms are used for two purposes: listing records on screen and printing reports. This chapter explains the following operations:

- Using the Form Wizard to create output forms for listing records on-screen,
- Using the Form Wizard to create forms for output forms and printed reports,
- Using the Form editor to customize output forms and printed reports.

In numerous cases, you can create a report more quickly using the Quick Report editor. However, the Form editor gives you greater control over the final appearance of your report. For more information about using the Quick Report editor, refer to the *4th Dimension User Reference*.

Output Forms for Listing Records

A form that displays records as a columnar list contains separate areas:

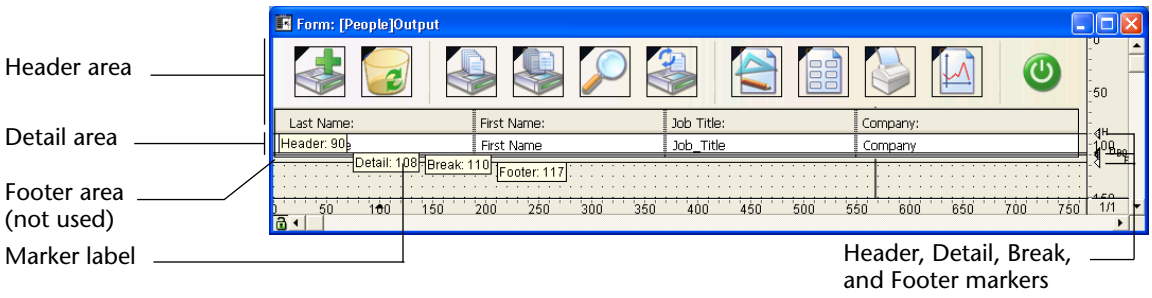
- **Header area** contains the report title, column headers, and form management buttons or objects (in custom applications only),
- **Detail area** contains the body of the report,
- **Footer area** contains buttons or summary calculations based on all the records in the report
- **Break area** contains text or graphics that appear after the list of records and summary calculations based on all the records or subgroups of records.

When you create a List form using the Form Wizard, it automatically creates these areas for you. It places the form title (the table name) and the field names in the Header area and a one or more buttons in the Footer area[1]. The fields you select are placed in a row in the Detail area. A small Break area is created but the Form Wizard puts nothing in it.

When you open the form in the Form editor, you can modify the size of each of these areas, modify the contents of any area, add objects to the Break area, and create additional Break areas for summary calculations.

The areas of the form that function as the Header, Detail, Break, and Footer areas are controlled by *output control lines*. By dragging the output control lines vertically, you can change the size of each area.

The following illustration shows an output form that was created using the Basic screen of the Form Wizard (XP template without labels).

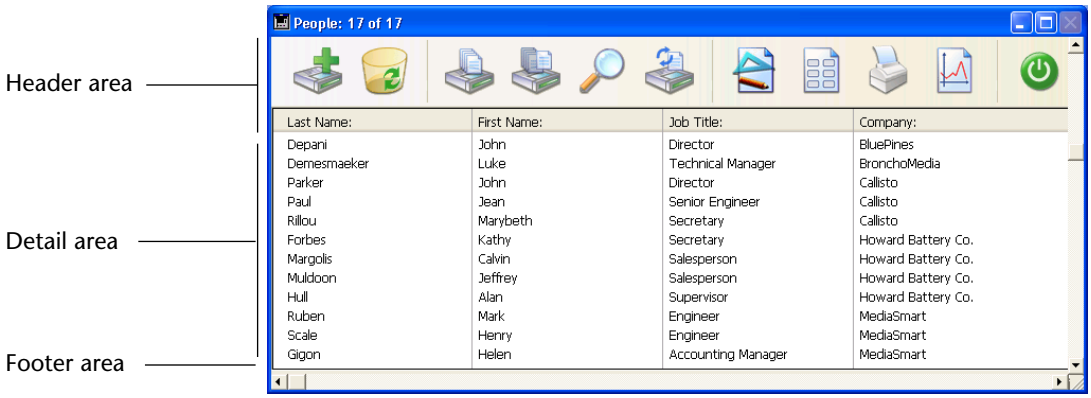


Note You can choose to hide/display markers and their labels. For more information, refer to "Showing/Hiding Elements in the Form Editor" on page 270.

The dotted horizontal lines divide the report into Header, Detail, Break, and Footer areas. The area from the top of the form to the Header line is the Header area. Similarly, the area between the Header and Detail lines is the Detail area, and the Footer area extends from the top of the Break line[2] (labelled B0) to the Footer line.
You adjust the sizes of each area by dragging the Header, Detail, Break, or Footer markers or their labels vertically.

1. The exact contents of the Header and Footer areas depend on the specific Form Wizard options that you select.
2. For more information on Break areas refer to "Reports with Breaks" on page 497.

When this form is used in the Custom Menus environment or a custom application, it looks like this:



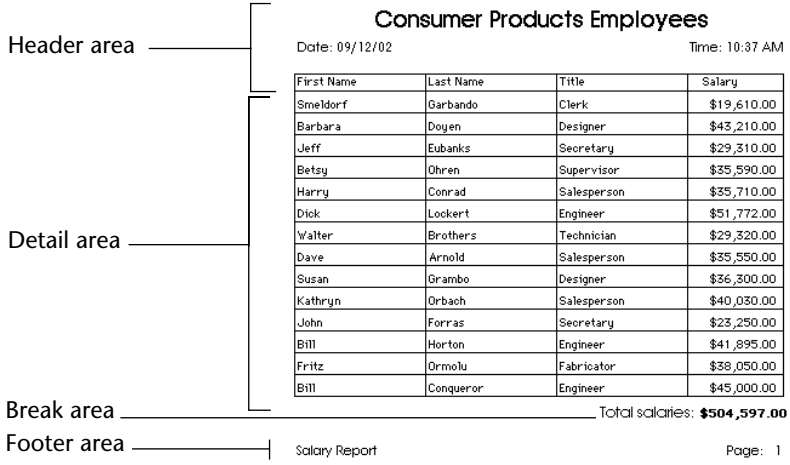
The Detail area expands dynamically as the window is resized, while the Header and Footer areas remain a fixed size.

These areas work slightly differently in the User and Custom Menu environments. For more information, refer to the next section, "Output Control Lines" on page 480.

In a form that lists records on screen, the Header and Footer areas can include clickable and non-enterable objects such as buttons, radio buttons, hierarchical lists, and so on. For more information, refer to "Output Control Lines" on page 480.

In a printed report, a Header area often contains the date, the time, and a running title as well as column labels. Records appear in the Detail area. A calculated total may appear in the Break area. The Footer area contains the page number.

The following illustration identifies the different areas as they appear in a printed report.



A report may have additional Break areas for subtotals and other calculations. A report may also have additional Header areas that appear within the body of the report. The additional Header areas are used to identify subgroups. For an example of a report with several Header and Break areas, see the section "Creating Additional Control Lines" on page 499.

Output Control Lines

You control the Header, Detail, Break, and Footer areas with the output control lines in the Form editor. You move the control lines vertically to allow more or less space for each area. Any object that you place in these areas is displayed or printed at the appropriate location.

The following explains how these areas work when the form is displayed or printed:

- **Header area** The Header area is displayed at the top of each screen in the User and Custom Menus environments and is printed at the top of each page of a report. The Header area is defined as the area above the Header control line (H). You make the Header area smaller or larger by dragging the Header control marker vertically. You can use the Header area for column names, for instructions, additional information, or even a graphic such as a company logo or a decorative pattern.

You can also place active objects in the Header area of forms displayed using the DISPLAY SELECTION and MODIFY SELECTION commands.

Note This type of operation does not apply to forms used in the User environment or to subforms.

Only objects that are clickable and non-enterable can be inserted. This includes:

- buttons, 3D buttons, highlight buttons, picture buttons,
- pop-up menus/drop-down lists, picture pop-up menus and hierarchical pop-up menus,
- scrollable areas, hierarchical lists,
- radio buttons, 3D radio buttons, picture radio buttons,
- check box, 3D check box,
- thermometer, rulers, dials.

Note Combo boxes cannot be inserted, since they are enterable objects.

Standard actions such as **Validate**, **Cancel,** or **Automatic splitters** can be assigned to the inserted buttons. The following events apply to the active objects you insert in the header area: On Load, On Clicked, On Header, On Printing Footer, On Double Clicked, On Drop, On Drag Over, On Unload. For more information on the MODIFY SELECTION and DISPLAY SELECTION commands, refer to the *4th Dimension Language Reference* manual.

- **Detail area** The Detail area is displayed on the screen once for each record in the User and Custom Menus environments and is printed once for each record in a report. The Detail area is defined as the area between the Header control line and the Detail control line (D). You make this area smaller or larger by dragging the Detail control marker vertically. Whatever you place in the Detail area is displayed or printed once for each record. Most often you place fields or variables in the Detail area so that the information in each record is displayed or printed, but you can place other elements in the Detail area as well.

- **Break areas** Break areas are displayed once at the end of the list of records in the User and Custom Menus environments and are printed once after the records have been printed in a report. In the report above, the Break area is defined as the area between the Detail control line and the Break control line (labeled B0). There can be other Break areas in your report. You make Break areas smaller or larger by dragging the Break control marker vertically. You can use a Break area to display information that is not part of the records (instructions, current date, current time, etc.) or to display a line or other graphic element that concludes the screen display. In a printed report, you can use a Break area for calculating and printing totals and other summary calculations.

- **Footer area** The Footer area is displayed on screen only in the Custom Menus environment and in custom applications. It is always printed at the bottom of every page of a report. The Footer area is defined as the area between the Break control line (B0) and the Footer control line (F). You make this area smaller or larger by dragging the Footer control marker vertically. You can use the Footer area to print graphics, page numbers, the current date, or any text you want at the bottom of each page of a report. For output forms designed for use on screen, the Footer area typically contains buttons that give the user options such as doing a search or sort, printing records, or putting away the current report.

  Whenever any form is used for output, either for screen display or printing, the output control lines take effect and the areas display or print at designated locations. The output control lines also take effect when a form is used as the List form in a subform area.

  The output control lines have no effect when a form is used for input.

  Methods that are associated with objects in these areas are executed when the areas are printed or displayed as long as the appropriate events have been activated. For example, a object method placed in the Header area is executed when the On Printing Header event takes place.

  You can create additional control lines to define additional Break areas and Header areas for a report. These additional areas allow you to print subtotals and other calculations in a report and to display other information effectively. Additional control lines are discussed in the section "Reports with Breaks" on page 497.

  Working with the output control lines and the areas they define are described in detail in subsequent sections of this chapter.

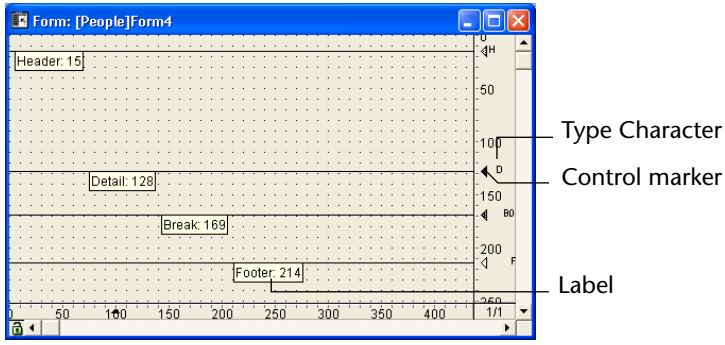
Moving Output Control Lines

You adjust the size of the Header, Detail, Break, and Footer areas by moving the output control markers.

Output control lines are displayed as dotted lines across the form. Each control line has an identifying marker and label that is displayed in the ruler. The control marker is the triangle in the ruler and the label is the letter or letters next to the marker. There is an additional label that you can display, it display's each control line's position and type. When you move a control line, the label displays its new position in real time.

You can also permanently display these marker labels, for more information, refer to "Showing/Hiding Elements in the Form Editor" on page 270. Labels allow you to move control lines even when the rulers are not displayed.

The figure below identifies control markers and labels:



To move a control line, drag the control marker or the marker label vertically.

Holding down the **Shift** key while dragging a control marker moves all control lines below that control marker. For example, to drag all control lines together, hold down **Shift** and drag the Header marker. To move all control lines except the Header control line, **Shift+drag** the Detail marker.

The control lines cannot be dragged out of order. For example, if you attempt to drag a Footer control line higher than a Break control line, the drag operation automatically stops when the Footer marker reaches the Break marker.

You can place markers and control lines on top of one another. Placing one marker on top of another reduces its area to nothing, removing it from the report. For example, if you have nothing to print in a Break area, you can drag the Break marker on top of the Detail marker. Doing so prevents 4th Dimension from creating space for a Break area. The report can thus utilize all the space available on the page.

If you don't want to print any details, drag the Detail marker on top of the Header marker. If you don't need a Header, drag the Header marker to the very top of the form (at point 0).

Creating Output Forms

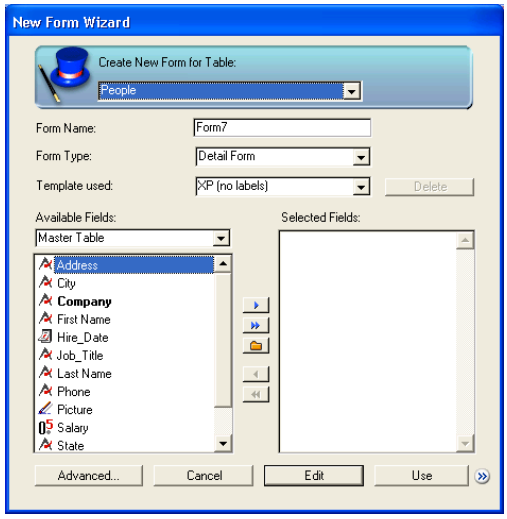
You use the Form Wizard to create an output form for each table in your database. The principle is similar to that used for creating input forms. You simply select a different form type. If you use the Wizard's Advanced screen, a different set of options is available.

As with input forms, you can create an output form using either the Basic or Advanced screens of the Form Wizard.

► To create an output form:

1 **Choose New Form from the Design menu.**
OR
Select the master table for the form in the Forms page of the Explorer and click New.

The Basic screen of the Form Wizard appears.



2 If you want to preview the effects of your settings, expand the preview area by clicking the Preview Area icon.

3 If necessary, use the "New Form for Table:" drop-down list to select the master table for the form.

The Fields list changes to reflect your selection.

4 Name the form by filling in a name in the "Form Name" area.

You can refer to the form by name using the language.

5 Choose List form from the "Form Type" drop-down list.

This selection instructs the Wizard to place the output control lines in the correct locations for lists, place the fields in a row in the Detail area, and add buttons to the Footer area (for use in custom applications only).

6 Choose a template that is appropriate for output forms.

The template controls several aspects of the appearance of the form, including font attributes, field label placement, the design of decorative rectangles surrounding fields, and platform interface. 4th Dimension ships with several templates. You can also create custom templates with the Form Wizard and add them to this list. For more information about adding custom templates, see the section "Creating a Form Template" on page 258.

7 **Select the fields you want on your form.**
For complete information about selecting fields for the form, refer to the section, "Selecting Fields for the Form" on page 237.

8 **If you want to generate the new form and modify it in the Form editor, click Edit.**
OR
If you want to generate the form, set it as the current form and switch to the User environment, click Use.
OR
If you want to customize the new form with the Forms Wizard's advanced options, click Advanced.

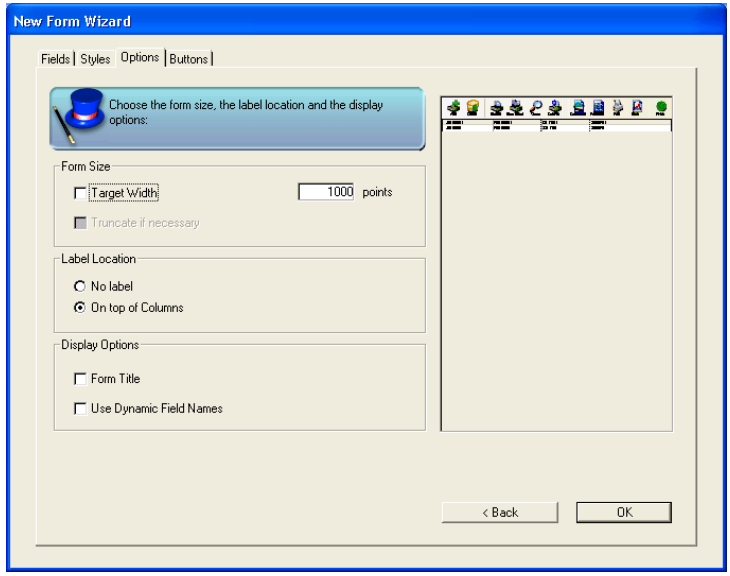
Using the Form Wizard's Advanced Options

When you create a List form, the Advanced screen of the Form Wizard has the following pages:

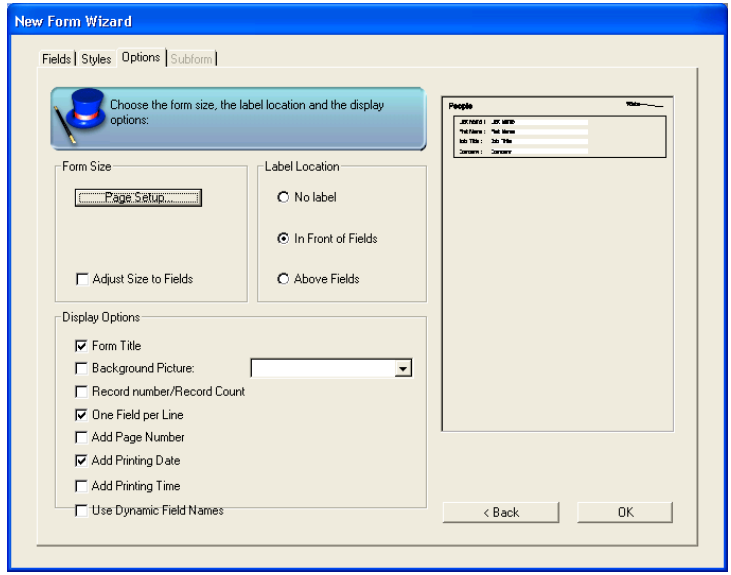
- **Fields** Similar to the Fields page for Detail forms, except that grouping fields is not relevant for List forms and is not available.
- **Styles** Same as the Styles page for Detail forms.
- **Options** Similar to the Options page for Detail form type, with the following differences.
  - In the Detail and List Form for Printing form type, the Form Size area has a Page Setup button. This button lets you choose the page size for the printed report. When you choose the page size, 4th Dimension adjusts the size of the form and preview area.
  - In the List form type, the Form Size area has a Target Width area. When you use these options, 4th Dimension tries to fit the fields in a particular width. For more information, see the section "Using the Target Width Option" on page 489.

- In the Display Options area, you can add variables to printed reports that display the page number, printing date, and printing time.

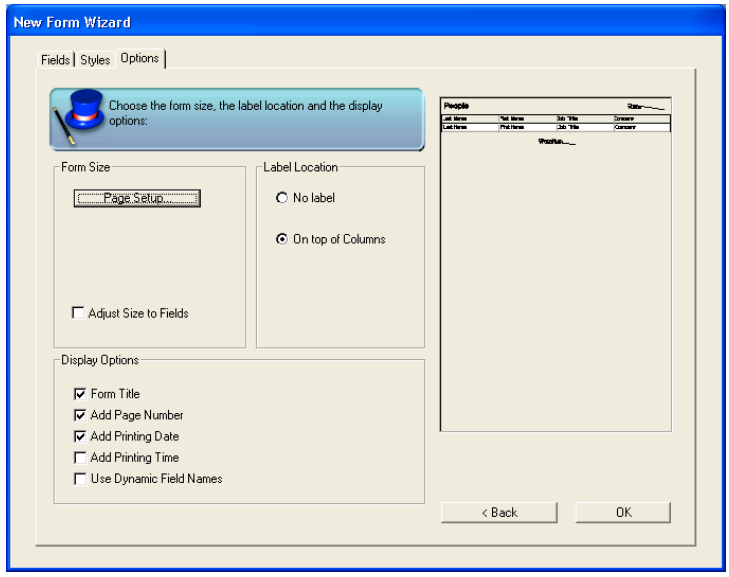
List Form



Detail Form for Printing

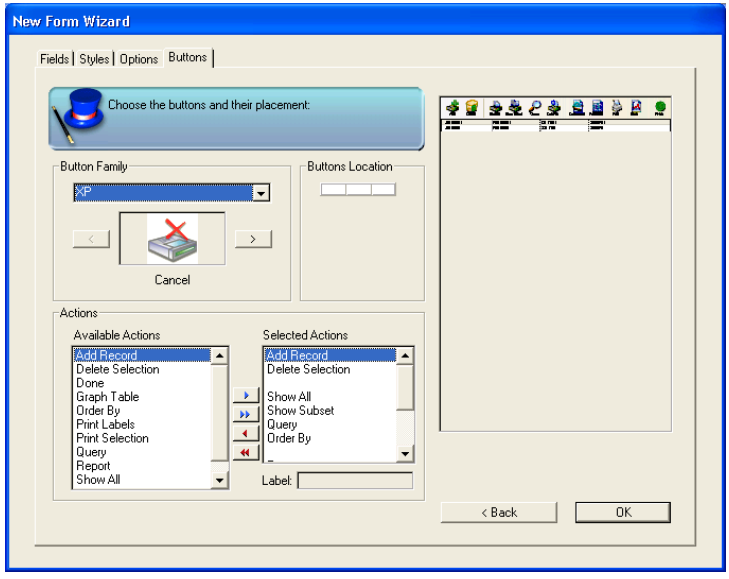


List Form for Printing



- **Buttons** Similar to the Buttons page for Detail forms, except that different buttons are available. The Buttons page is not available for the List and Detail forms for Printing form types.

List Form



The buttons are not automatic in the same sense as buttons for input forms; instead, 4th Dimension automatically creates an object method for each button. Using the Form editor, you can edit the object methods.

For more information on working with the Buttons page, see the section "Customizing Buttons on the Form" on page 250.

Using the Target Width Option

When you create a List form for screen display, you can use an Advanced option to tell the Form Wizard to take a target width into account when it creates the form. If you do not use this option, the width of the output form will be based on the sum of the field lengths of the fields that you add to the form.

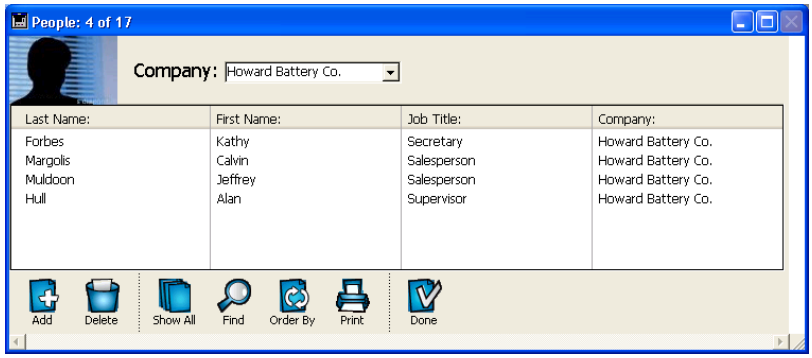
If you click Target Width and enter a target width in pixels, the Form Wizard will try to fit all the fields on the form by reducing the widths of the fields. If you also click Truncate if Necessary, the Form Wizard will also remove one or more fields to make the form width less than or equal to the target width. If you don't check Truncate if Necessary, the form's width may be slightly larger than the target width.

Modifying an Output Form in the Form Editor

A List form that is created using the Form Wizard works well as a listing of records. If needed, you can make the following simple modifications:

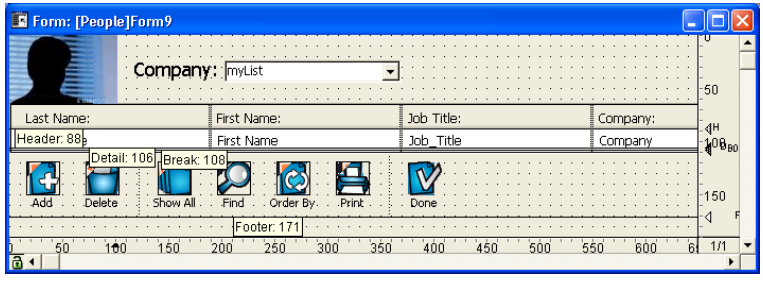
- Replace fields with variables and add methods,
- Use the platform interface, appearance, font attributes, fill, border, or color options to modify individual objects on the form,
- Change the widths of the fields or variables,
- Move the output control lines,
- Add a custom graphic in the Header area,
- Add variables in the Header area (for List forms designed to be displayed on screen in custom applications — refer to "Output Control Lines" on page 480).

The following is a custom output form being used for managing personnel :



The control buttons have been placed in the footer area and a customized picture has been placed in the header area.
The header area also includes a drop-down list that lets you select the company whose employees you want to display (the form is displayed using the DISPLAY SELECTION or MODIFY SELECTION command).

Here is what the design for this form looks like in the Form editor:



The control lines can be moved, for example, in order to adapt to the size of the picture inserted into the header area.

Displaying Several Lines Per Record

You may want to display fields on more than one line. 4th Dimension allows you to use several lines for each record. Expand the Detail area so that more than one line appears in this area. When you use several lines for a single record, graphic elements such as lines and boxes can be useful to separate fields and records.

Here is an example of an output form for a phone message management database which uses two lines per record:

Modifying Output Forms for Reports

Printed reports, unlike screen display forms, can make use of the Break area at the end of the report. For material that appears at the bottom of each page, printed reports use the Footer area.

This section provides suggestions and directions specifically for designing forms to be used for printing. It covers lists, reports that print one record per page, expandable areas for printing invoices and variable text fields, form letters, and custom mailing labels.

Printed Columnar Reports

Lists that display columns of information are common in printed reports. You might publish lists of telephone numbers, prices, results, specifications, or parts.

When you create a columnar report using the Form Wizard, you choose **List Form for Printing** as the Form Type.

Like the lists you design to display records on the screen, a printed list presents columns of information, can include column headings above each column, and can use graphic elements to enhance or clarify the report.

The Break area, which is printed once at the end of the report, is used for calculating totals. For a discussion of using methods for calculating totals, see the section “An Example Report” on page 506.

One Record Per Page Reports

You may need to print one record per page. For example, you may want to use an invoicing database to print a copy of each invoice for your records.

When you need to create such a report, you choose **Detail Form for Printing** as the Form Type in the Form Wizard.

Place the Header (H) control line at the top of the page and arrange the fields and other report elements below it.

If your form displays records in a subform, be sure that the subform is set to print with fixed frame so that the records do not wrap onto additional pages. For information about printing with fixed frame, see the section "Printing Subforms, Pictures, and Text Fields" on page 509.

Drag the Detail (D), Break (B0), and Footer (F) control lines to the bottom of the page to ensure that only one record is printed per page.

Using Subforms

You are probably very familiar with an invoice. A typical invoice shows a name and billing address, a shipping address, a series of items or services purchased, and a total.

An invoicing database includes an output form for printing full-page invoices. For example, an invoice draws information from two tables: an [Orders] table that provides the customer information (bill to and ship to addresses) and a [Line Items] table that provides the line items. The total for the order is calculated and kept in the [Orders] table.

The form for an invoice is created in the [Orders] table and uses a subform area for the line items. The subform area can expand during printing to print all the line items, even if the invoice requires a second page. For more information, see the section "Printing Subforms, Pictures, and Text Fields" on page 509.

The Detail area is expanded to the full-page size. One invoice is printed for each sales order, but as many line items as necessary are printed in the line items area of the invoice.

Report with a Text Field

Many databases allow the user to enter notes or descriptions in a text field. That same text field can be printed as an expandable area in the Body area.

Text fields can be expanded during printing to accommodate all of the data. For more information about printing text fields, see the section "Printing Subforms, Pictures, and Text Fields" on page 509.

Custom Mailing Labels

If you want to create special mailing labels, you can design a custom output form for them. The design can use graphic elements, any available fonts, and variables.

Tip It is often quicker and easier to create mailing labels using the Label Wizard in the User environment. For more information about the Label Wizard, refer to the *4th Dimension User Reference*.

The creation of output forms used for printing mailing labels is detailed in the paragraph "Printing Labels," page 512.

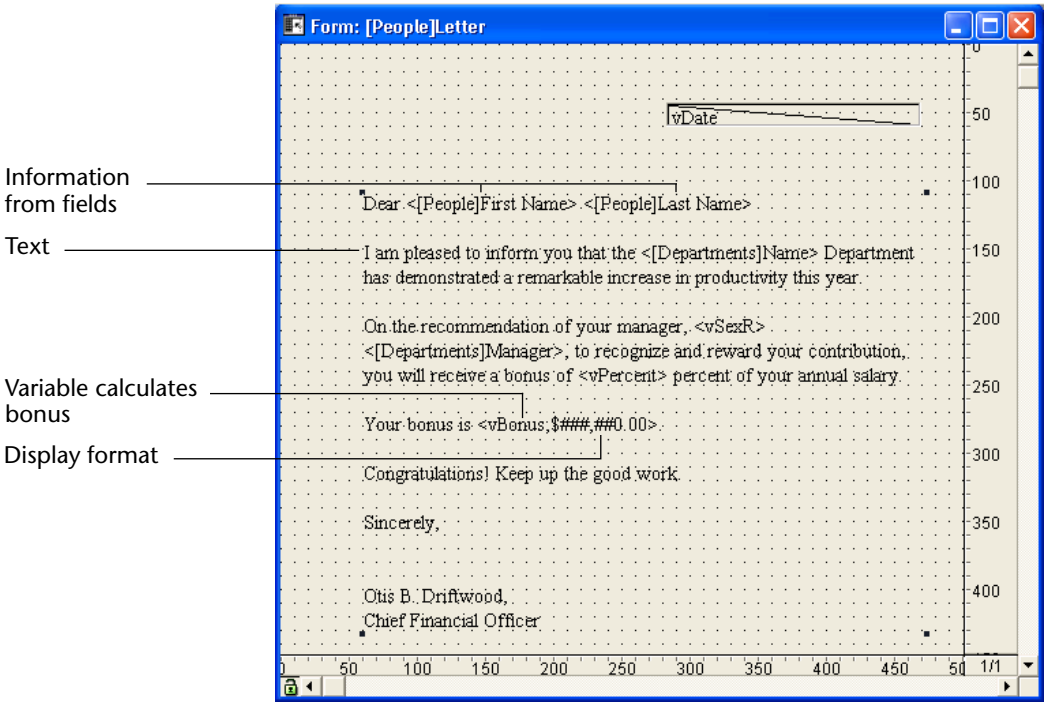
Creating Mail-Merge Documents

You can handle mail-merge tasks using an output form that embeds fields, field or table labels, or variables in a static text area. You can create documents that are the same in every respect except for names, addresses, and any special calculations you want to perform.

Mail-merge documents may be useful when you want to announce a special offer or inform your customers or clients of a business development (such as a move to a new location or a significant personnel change). Another typical use of mail-merge is to inform people that their account is due. You can create a variable and a method to calculate the exact amount.

When you create a form that does mail-merge, use the **Detail Form for Printing** option in the Form Type drop-down list in the Form Wizard. In the Form editor, create a text area that will contain both the static text and the fields, variables, or table or field labels that will change for each record. You then embed fields or variables in the text area. During printing, values from the fields or variables are inserted in the text.

The figure below shows fields and variables placed in a text area.



► To create an output form for mail-merge:

1 **In the Form Wizard, create a new form using the Detail Form for Printing option in the Form Type drop-down list.**

2 **Click Edit to open the form in the Form editor.**

3 **Select the Text Area tool and create a text area in the Detail area.**

4 **Type the text you want in your form letter, placing field and variable names between less than (<) and greater than (>) symbols where you want information from fields or variables.**

You can use a field from any table in the database. Fields from the master table do not have to specify the table name; they can be entered like this: <*field*>. Fields from other tables must specify the table name; they are entered like this: <*[tablename] field*>. When the form is printed, the information from the field for each record replaces the <*field*> element in the text area.

To insert table labels, enter: *<?[Table Name]>* or *<?[N]>* where N is the creation order for the table. To insert field labels, enter: *<?[Table Name]Field Name>* or *<?[X]Y>* where *X* is the creation order for the table and *Y* is the creation order for the field or *<?Y>* to insert a field of the current table. For more information, refer to "Inserting Dynamic Table and Field Names" on page 370.

A variable must be assigned a value in an object or form method.

You can specify how an embedded field or variable is displayed by inserting a semicolon followed by a display format directly after the field or variable name. For example, the mail-merge document shown above includes a display format for the vBonus variable. The embedded variable <vBonus;$###,##0.00> calculates the amount of the person's bonus and displays it in a dollar format. For more information about display formats, see the section "Display Formats" on page 396.

4th Dimension provides a shortcut for inserting fields in a text area. The shortcut allows you to choose the field name from a drop-down menu.

▶ To insert a field using the shortcut:

1 **Click to place the insertion point where you want the field.**

2 **Position the pointer in the text area and hold down the Alt key (Windows) or Option key (MacOS) while you press and hold down the mouse button.**

4th Dimension displays a pop-up menu of fields from the master table from which you can choose the field you want.

To choose a field from another table in the database, hold down Shift+Alt (Windows) or Shift-Command (MacOS) while you press and hold down the mouse button.

4th Dimension displays a hierarchical menu of tables and fields in the database. You can choose the table and field you want.

4th Dimension places *<field>* or *<[table]field>* in the text area at the insertion point.

When the report is printed, the values of the fields and variables embedded in the letter appear.

Basic Steps for Creating a Printed Report

You can perform most or all of these steps depending on the nature of the report you want to print. The actual order of steps is not critical. Typically, you design a report, preview it on screen, and then return to the form to make adjustments. The order given here is typical:

1 **Create a form using the Form Wizard.**

For information about creating a form, refer to Chapter 4.

2 **Move the control lines so that you will have enough space to place the various elements of the report.**

For information about moving control lines, see the section "Moving Output Control Lines" on page 483.

3 **Move elements into position.**

You can place fields, text, non-enterable objects, and graphic objects. Take time to align the objects to one another and to check the justification of data to be displayed in fields and active objects.

4th Dimension displays page size guidelines in the form so that you can place elements for printing.

For information about placing fields and active objects in a form, refer to Chapter 6.

4 **Adjust the control lines if necessary.**

Working with the relationship between control lines and elements to be printed, you can create the right amount of space for printing the Header, Detail, Break, and Footer elements.

5 **If necessary, create methods to calculate values, concatenate strings of characters, print additional text, and display the date, time, and page numbers.**

For information about creating methods, refer to "Using Object Methods with Fields and Objects" on page 462. For complete information about using object methods, refer to the *4th Dimension Language Reference* manual.

For information about calculating subtotals and other summaries (averages, counts, etc.), refer to "Reports with Breaks" on page 497.

Be sure that you activate the appropriate events in the Events section of the object properties for each method you use; otherwise the method will not execute during printing.

6 **Go to the User environment to test the report.**

7 **Create the selection of records you want to use for the report.**

For information about creating a selection of records, refer to the *4th Dimension User Reference*.

8 **Sort the records according to how you have designed the report.**

For information about sorting records, refer to the *4th Dimension User Reference*.

9 **Preview the report by printing it to the screen.**

The report form uses the Page Setup settings that were in effect when the form was created in the Design environment. If you make any changes to the Page Setup dialog box, be sure to make the same changes in the Design environment.

If the report form needs to be adjusted, return to the Design environment to make any necessary changes.

10 **Print the report.**

For information about printing in the User environment, refer to the *4th Dimension User Reference*.

Reports with Breaks

4th Dimension can print reports that work with Breaks and Break Headers. A Break is created when you sort the records.

Suppose you have a collection of compact discs that you keep track of in a 4th Dimension database and you want to print a list that arranges the information by artist. When you sort the records by artist, all the records fall into distinct groups. The "Break" occurs after the last record in each group is printed.

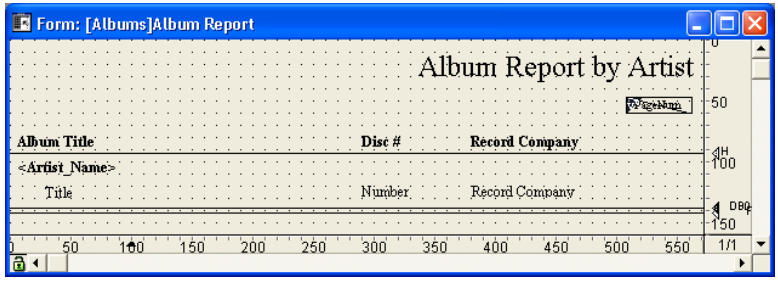
Here is how the report looks when previewed to screen:

Album Report by Artist

Page 4

Album Title	Disc #	Record Company
THE PLANETS	023	Deutsche Grammophon
THE PLANETS, SUITE DE BALLET, OP. 10	250	Enigma Classics
Harry Belafonte		
ALL TIME GREATEST HITS VOL.1	025	BMG Music
PARADISE IN GAZANKULU	028	Capitol Records
Hector Berlioz		
SYMPHONIE FANTASTIQUE	192	Enigma Records
Hiroshima		
ANOTHER PLACE	036	
Huey Lewis And The News		
FORE!	031	Chrysalis Records
INXS		
KICK	203	Atlantic Recording Corporation
Jean-Luc Ponty		
COSMIC MESSENGER	029	Atlantic Recording Corporation
Jerry Goodman		
ARIEL	073	Private Music, Inc.
Joaquin Rodrigo		
CONCIERTO DE ARANJUEZ	233	Enigma Classics
Joe Sample		
SPELLBOUND	179	Warner Bros. Records Inc.
Johann Sebastian Bach		
BACHBUSTERS	080	TELARC DIGITAL
BRANDENBURG CONCERTOS NOS. 1, 2, & 3	125	Enigma Records
BRANDENBURG CONCERTOS NOS. 4, 5 & 6	163	Enigma Records
ORGAN FAVOURITES	248	Enigma Classics
Johann Strauss, Jr.		
STRAUSS FESTIVAL VOLUME 1: FAMOUS WALTZES, PC	096	Enigma Records
STRAUSS FESTIVAL VOLUME 2	159	Enigma Records
Johannes Brahms		
HUNGARIAN DANCES NOS. 1-21	243	Enigma Classics
HUNGARIAN DANCES NOS. 1-21	146	Enigma Records
SYMPHONY NO.4 ACADEMIC FESTIVAL OVERTURE	127	Enigma Records
Joni Mitchell		
COURT AND SPARK	108	Asylum Records
Joseph Haydn		

4th Dimension provides features that you can use to display the information attractively. Here is this form in the Design environment.



In order to generate a report that uses Break levels and Headers, you must first initiate Break processing. For more information about the methods you can use to initiate Break processing, refer to "Initiating Break Processing" on page 501.

A Break Header is printed once before the group of records it refers to and a Break is printed once after the group of records it refers to. In the illustration on the previous page, the Break is called a "level 1 Break" and the Break Header is called a "level 1 Break Header," because the Break occurs as a result of the first sort field.

You can use up to nine break levels. If you use Subtotal to initiate Break processing, you need to sort on one more field than the number of Breaks you use. In this case, if you use one Break level, you must sort on two fields. If you use three Break levels, you must sort on four fields.

This section explains how to create reports using Breaks. A full-scale example is provided at the end of this section. If you have any trouble understanding anything in this section, try reading the whole section and then coming back to the trouble spot. The concepts are interrelated and are often easier to understand in context than by themselves.

Creating Additional Control Lines

The report examples shown in this section use Break levels and Break Headers. To create areas for these features, you create additional control lines.

The Form editor always starts with the original control lines, labeled H, D, B0, and F. B0 stands for "Break at level 0." Level zero takes in all the records; it occurs after all the records are printed. Additional Break control lines are designated with numbers. A control line labeled B1 stands for "Break at level 1." A level one Break occurs after the records grouped by the first sort field are printed.

Label	Explanation	Prints after groups created by:
B1	Break at level 1	First sort field
B2	Break at level 2	Second sort field
B3	Break at level 3	Third sort field

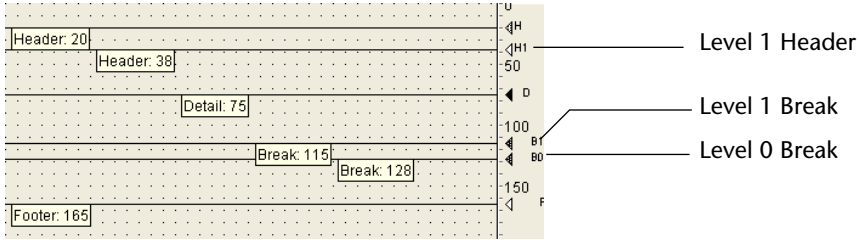
H stands for "Header," which is printed at the top of each page. Additional Header control lines are associated with Breaks. H1 stands for "Header at level 1." A level 1 Header is printed just before the records grouped by the first sort field are printed.

Label	Explanation	Prints before groups created by:
H1	Header at level 1	First sort field
H2	Header at level 2	Second sort field
H3	Header at level 3	Third sort field

You create additional control lines by holding down the **Alt** key (Windows) or **Option** key (MacOS) while clicking the appropriate control marker. You use a Break control line to create a Break area for the designated level. You use a Break Header control line to create a Break Header area at the designated level.

If you use the Subtotal function to initiate Break processing, you should create a Break area for every level of Break that will be generated by the sort order, minus one. If you do not need anything printed in one of the Break areas, you can reduce its size to nothing by placing its marker on top of another control line. If you have more sort levels than Break areas, the last Break area will be repeated during printing.

The figure below shows additional control lines.



► To create a new Break or Break Header control lines:

1 **Hold down the Alt key (Windows) or Option key (MacOS) and click any Break (B) or Break Header (H) marker.**

4th Dimension creates a new Break or Break Header control line.

The new line is positioned behind the existing control line; to see the new control line, you need to drag the existing line off of it.

2 **Drag the existing line away from the new line.**

3 **Position both lines where you want them.**

► To delete Break or Break Header control lines that you have created:

1 **Hold down the Ctrl key (Windows) or Command key (MacOS).**

2 **Click on the Break, Break Header, or label of the control line that you want to delete.**

4th Dimension deletes the control line and, if necessary, renumbers the remaining lines.

You cannot delete the original control lines H, D, B0, and F.

Initiating Break Processing

To allow 4th Dimension to print Break Header areas, accumulate subtotals, and perform other aspects of Break processing, you must first initiate Break processing in the report form. You initiate Break processing by either:

- Placing the Subtotal function in an object or form method,
- Executing the BREAK LEVEL and ACCUMULATE commands before printing the report.

If the database will be used in interpreted mode only, you can use whichever method you prefer. However, you must use BREAK LEVEL and ACCUMULATE to initiate Break processing if the database will be used in compiled mode. In addition, if you use the Subtotal function, you must also sort the records on one more sort field than the number of Break levels you use. For example, if you use two levels of Breaks in your report, you must sort on three fields.

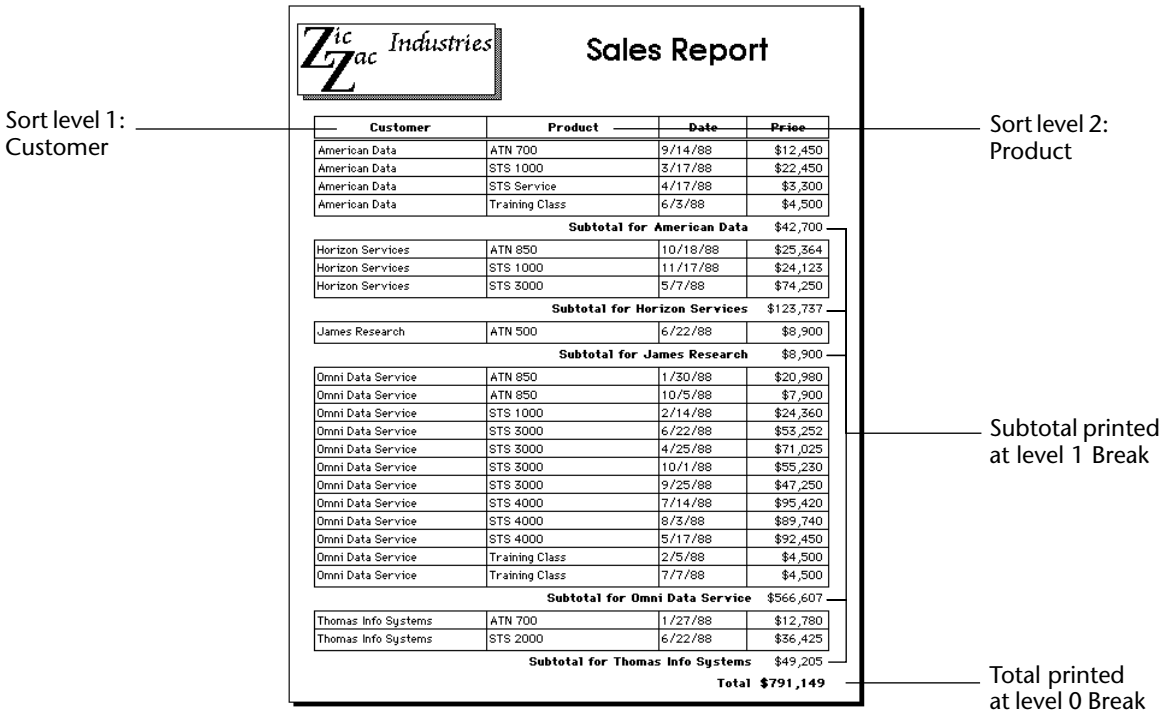
For more information about initiating Break processing, refer to the *4th Dimension Language Reference* manual.

Reports with Subtotals

This section describes in detail how the sort order affects reports and how to use additional Break areas for creating subtotals.

4th Dimension can automatically calculate and print totals and subtotals.

The figure below shows a report that calculates subtotals for each customer and a total at the end of the report.

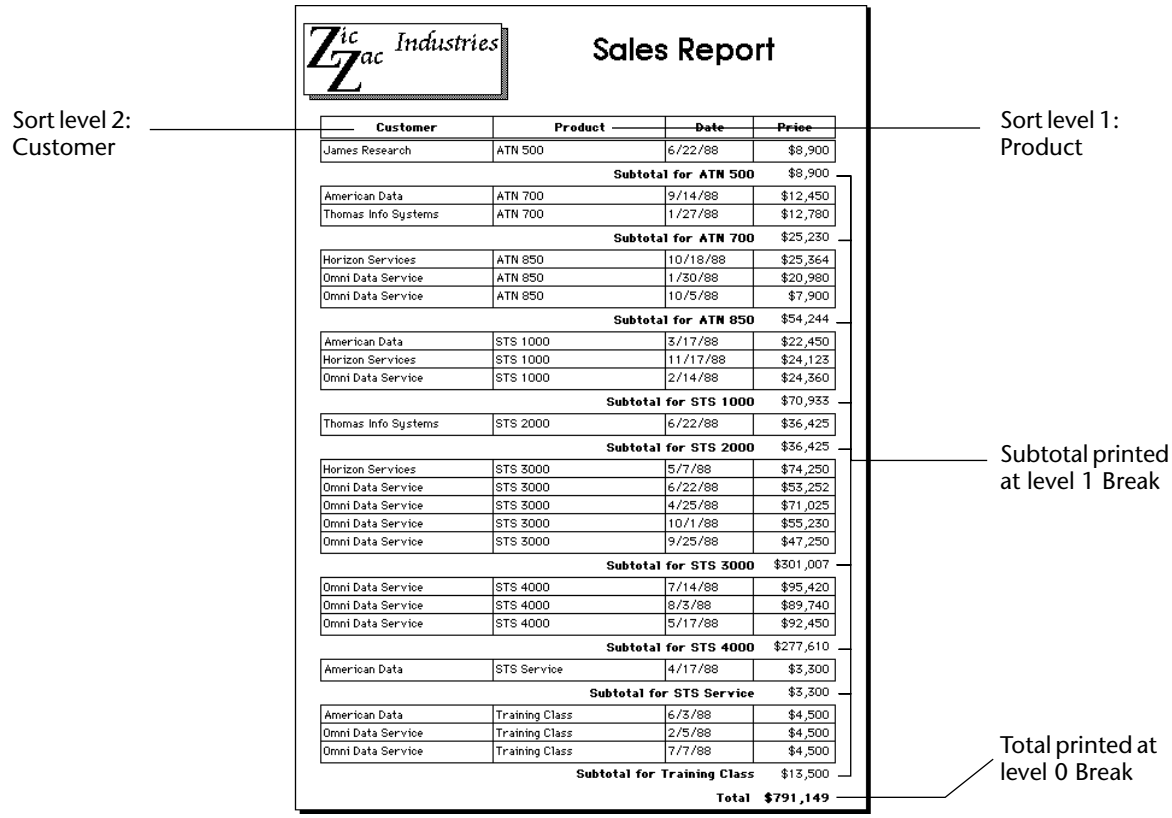


These records have been sorted by customer and product. After the records for each customer have been printed, the subtotal for the customer is calculated and printed. After the records for the entire report have been printed, the total is calculated and printed.

As you can see in the figure above, the subtotals are calculated and printed after the records for each customer. 4th Dimension knows when to perform the calculation and print the subtotal because it has been instructed to do so when the value in the first sort field changes (where it “breaks”).

The customer subtotal is calculated in what is called a level 1 Break because it is based on a change in the first sort level (in this case, the Customer field). The grand total is calculated in a level 0 Break. A level 0 Break includes all the records and occurs at the end of the report.

The figure below shows another example of calculations during a Break, using the same records, but with a different sort order to create a different Break. This time the records have been sorted by product and customer. The subtotals are calculated when the value in the Product field changes. This is still a level 1 Break, but the Break is on a different field.



Zic Zac Industries

Sales Report

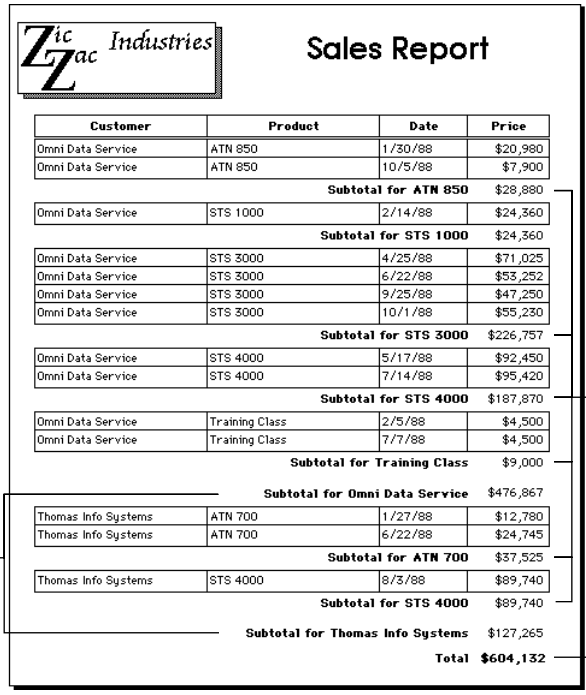
Customer	Product	Date	Price
James Research	ATN 500	6/22/88	$8,900
	Subtotal for ATN 500		$8,900
American Data	ATN 700	9/14/88	$12,450
Thomas Info Systems	ATN 700	1/27/88	$12,780
	Subtotal for ATN 700		$25,230
Horizon Services	ATN 850	10/18/88	$25,364
Omni Data Service	ATN 850	1/30/88	$20,980
Omni Data Service	ATN 850	10/5/88	$7,900
	Subtotal for ATN 850		$54,244
American Data	STS 1000	3/17/88	$22,450
Horizon Services	STS 1000	11/17/88	$24,123
Omni Data Service	STS 1000	2/14/88	$24,360
	Subtotal for STS 1000		$70,933
Thomas Info Systems	STS 2000	6/22/88	$36,425
	Subtotal for STS 2000		$36,425
Horizon Services	STS 3000	5/7/88	$74,250
Omni Data Service	STS 3000	6/22/88	$53,252
Omni Data Service	STS 3000	4/25/88	$71,025
Omni Data Service	STS 3000	10/1/88	$55,230
Omni Data Service	STS 3000	9/25/88	$47,250
	Subtotal for STS 3000		$301,007
Omni Data Service	STS 4000	7/14/88	$95,420
Omni Data Service	STS 4000	8/3/88	$89,740
Omni Data Service	STS 4000	5/17/88	$92,450
	Subtotal for STS 4000		$277,610
American Data	STS Service	4/17/88	$3,300
	Subtotal for STS Service		$3,300
American Data	Training Class	6/3/88	$4,500
Omni Data Service	Training Class	2/5/88	$4,500
Omni Data Service	Training Class	7/7/88	$4,500
	Subtotal for Training Class		$13,500
	Total		**$791,149**

Additional Break Levels

You can provide additional summary calculations by adding another sort level and another Break level.

The following figure shows sales records sorted by customer, product, and salesperson. Summary calculations show two sets of subtotals: one subtotal for each customer, and, within each customer, subtotals for each product for the customer. Finally, this report calculates a total for the entire company.

These are examples of calculations performed at level 2 Breaks, at level 1 Breaks, and at the level 0 Break.



Zic Zac Industries

Sales Report

Customer	Product	Date	Price
Omni Data Service	ATN 850	1/30/88	$20,980
Omni Data Service	ATN 850	10/5/88	$7,900
		Subtotal for ATN 850	$28,880
Omni Data Service	STS 1000	2/14/88	$24,360
		Subtotal for STS 1000	$24,360
Omni Data Service	STS 3000	4/25/88	$71,025
Omni Data Service	STS 3000	6/22/88	$53,252
Omni Data Service	STS 3000	9/25/88	$47,250
Omni Data Service	STS 3000	10/1/88	$55,230
		Subtotal for STS 3000	$226,757
Omni Data Service	STS 4000	5/17/88	$92,450
Omni Data Service	STS 4000	7/14/88	$95,420
		Subtotal for STS 4000	$187,870
Omni Data Service	Training Class	2/5/88	$4,500
Omni Data Service	Training Class	7/7/88	$4,500
		Subtotal for Training Class	$9,000
		Subtotal for Omni Data Service	$476,867
Thomas Info Systems	ATN 700	1/27/88	$12,780
Thomas Info Systems	ATN 700	6/22/88	$24,745
		Subtotal for ATN 700	$37,525
Thomas Info Systems	STS 4000	8/3/88	$89,740
		Subtotal for STS 4000	$89,740
		Subtotal for Thomas Info Systems	$127,265
		Total	**$604,132**

Subtotal printed at a level 2 Break

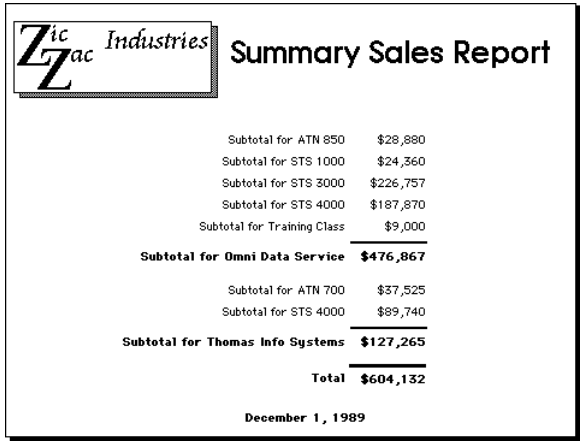
Subtotal printed at a level 1 Break

Total printed at a level 0 Break

The subtotal calculations are performed only for the group of records that precedes the Break. For example, a subtotal is calculated for each product sold to each customer. The subtotal for the customer is calculated for all products sold to that customer.

Summary Reports

You can create a report that prints only summary information. Such a report displays only the subtotals and totals with appropriate additional text. The following illustration shows a report with only summary information.



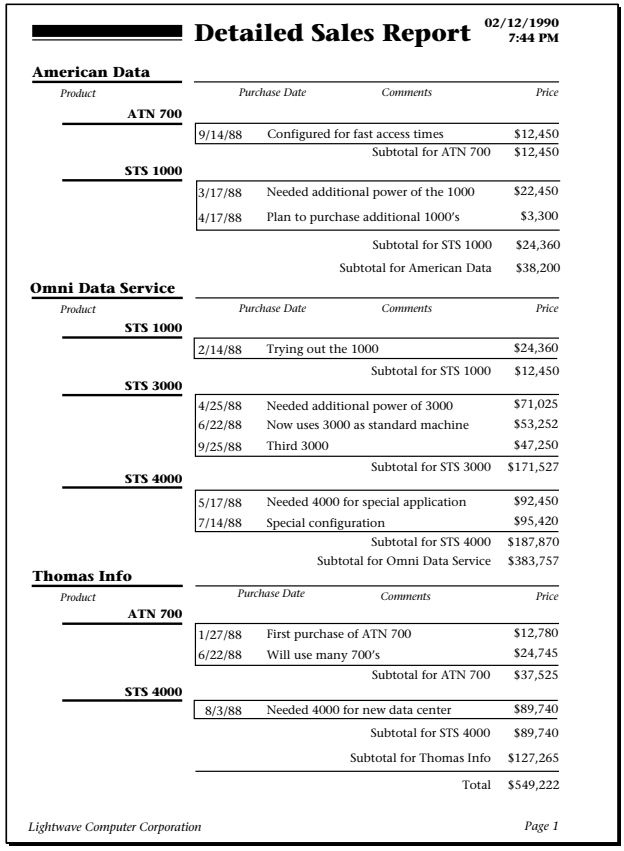
In this report, the records are sorted by customer, product, and date and the calculations are performed during the Breaks created by the sort order. The Detail area for each record is not printed; the records are used only to provide values for the calculations. Notice that 4th Dimension goes through the records from the first to the last during the printing of the report so that it can calculate these figures. (You create this kind of report by placing the Detail control line on top of the Header control line, leaving no space for details to print.)

You can ask 4th Dimension to perform additional calculations on a list including subtotals, averages, maximum and minimum values, page totals to be printed in a footer, and weighted averages. These calculations, and others, use 4th Dimension's summary functions. For information about how to use summary functions, refer to the *4th Dimension Language Reference* manual.

An Example Report

This section describes an example report and shows how the finished report is related to a form in the Form editor and to the methods that control the printing.

The following illustration shows a finished report.



Detailed Sales Report 02/12/1990 7:44 PM

American Data

Product	Purchase Date	Comments	Price
ATN 700			
	9/14/88	Configured for fast access times	$12,450
		Subtotal for ATN 700	$12,450
STS 1000			
	3/17/88	Needed additional power of the 1000	$22,450
	4/17/88	Plan to purchase additional 1000's	$3,300
		Subtotal for STS 1000	$24,360
		Subtotal for American Data	$38,200

Omni Data Service

Product	Purchase Date	Comments	Price
STS 1000			
	2/14/88	Trying out the 1000	$24,360
		Subtotal for STS 1000	$12,450
STS 3000			
	4/25/88	Needed additional power of 3000	$71,025
	6/22/88	Now uses 3000 as standard machine	$53,252
	9/25/88	Third 3000	$47,250
		Subtotal for STS 3000	$171,527
STS 4000			
	5/17/88	Needed 4000 for special application	$92,450
	7/14/88	Special configuration	$95,420
		Subtotal for STS 4000	$187,870
		Subtotal for Omni Data Service	$383,757

Thomas Info

Product	Purchase Date	Comments	Price
ATN 700			
	1/27/88	First purchase of ATN 700	$12,780
	6/22/88	Will use many 700's	$24,745
		Subtotal for ATN 700	$37,525
STS 4000			
	8/3/88	Needed 4000 for new data center	$89,740
		Subtotal for STS 4000	$89,740
		Subtotal for Thomas Info	$127,265
		Total	$549,222

Lightwave Computer Corporation *Page 1*

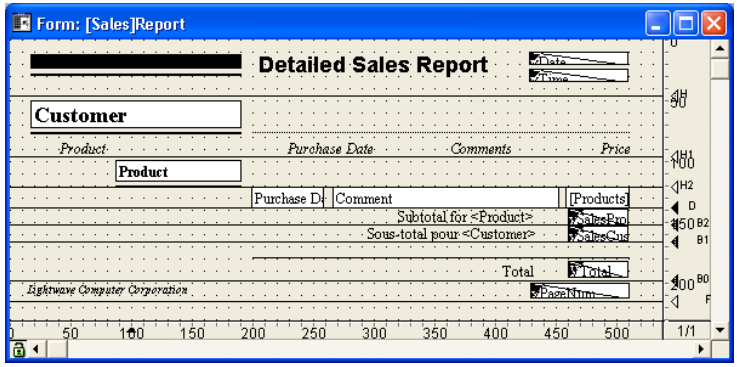
In this example, the page Header contains the date, the time, and the report title. The Break Headers contain the customer name and column headings for the information presented during the first Break. The Detail areas contain data drawn directly from the records. The level 2 Break areas contain subtotals for products for each customer. The level 1 Break areas contain subtotals for each customer and the level 0 Break area contains a total for the report. The Footer contains the page number.

The report is sorted on one more level than Break levels. In this report, the sort fields are Customer, Product, and Date.

4th Dimension requires one more sort level than Break level for the Break processing method used in this report.

The Report Form

The following illustration shows the report form that created the report shown on this page.



Each control line in the form defines the bottom of its area. Whatever is placed in the form is printed at the appropriate place in the report. The Header area contains the elements that will be printed at the top of each page, the Detail area contains the elements that will be printed for each record, and so on. The following table shows what each of these control lines means.

Label	Explanation	Effect
H	Header area	Printed once at the top of each page
H1	L1 Header area	Printed once before each level 1 Break
H2	L2 Header area	Printed once before each level 2 Break
D	Detail area	Printed once for each record in the selection
B2	L2 Break area	Printed once at each level 2 Break (when the value in the second Sort field changes)
B1	L1 Break area	Printed once at each level 1 Break (when the value in the first Sort field changes)
B0	L0 Break area	Printed once at the end of the report
F	Footer area	Printed once at the bottom of each page

You can have additional Break areas and additional Break Header areas.

The Report Object Methods

The non-enterable objects that are placed in the Header, Break, and Footer areas are controlled by object methods. This section describes the methods used to print values in these locations.

Note The following code can be used only in object and form methods. It cannot be used in project methods.

The date is drawn from the system date by placing a non-enterable object named vDate in the Header area with this method:

vDate:= **Current Date**

The time is drawn from the system clock by placing a non-enterable object named vTime in the Header area with this method:

vTime:= **Current Time**

The subtotal for sales in the level 2 Break area is calculated and displayed in an object named vSalesProd with the following method:

vSalesProd:= **Subtotal** (Sales)

The subtotal for sales in the level 1 Break area is calculated and displayed in an object named vSalesCust with the following method:

vSalesCust:= **Subtotal** (Sales)

The total for sales in the level 0 Break area is calculated and displayed in an object named vSalesTotal with the following method:

vSalesTotal:= **Subtotal** (Sales)

Note that even though all three objects use the same calculation, they produce different results. Because they are placed in different Break areas, they are executed at different times and perform their calculations for different groups of records. For an explanation of Break levels, see the section "Reports with Breaks" on page 497.

vPage:= "Page " + **String** (**Current form page**)

The Current form page function returns the page number.

Printing Subforms, Pictures, and Text Fields

You can use subforms, picture fields, and text fields in a report. These objects can be set to print with either a fixed or variable frame. Fixed frame objects print within the confines of the object as it was created on the form. Variable frame objects expand during printing to include the entire contents of the object.

You cannot place more than one variable frame object side-by-side on a form. You can place non-variable frame objects on either side of a variable frame Picture or Text field provided that the variable frame object is at least one line longer than the object beside it and that all objects are aligned to the top. Otherwise, during printing, the contents of the other fields will be repeated for every horizontal slice of the variable frame object.

You cannot place objects on either side of a variable frame subform.

You can place variable objects with the Body area only.

Printing Subform Areas

You usually use an output form to print records in a subform.

► To set the print option for a subform:

1 In the Form editor, display the Object properties palette or the Property List, then click on the subform.

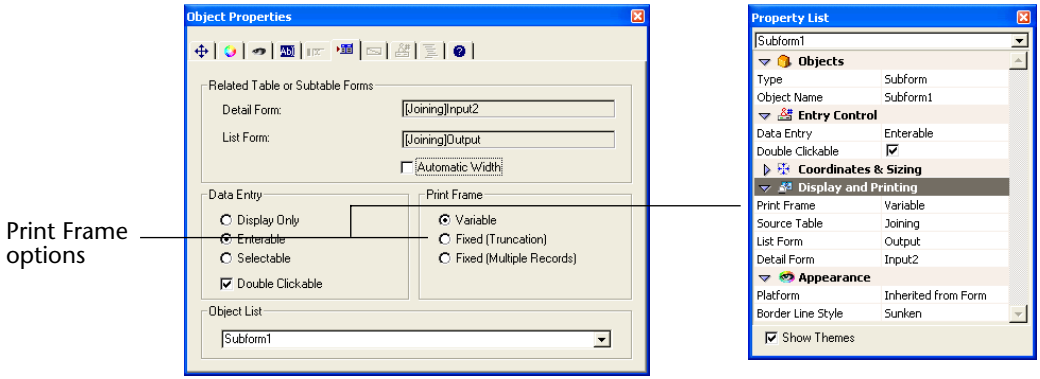
Note Double-clicking a subform causes it to open in a new window of the Form editor.

2 In the Object Properties window, click the Subform tab and select a print option as shown below.
OR
In the Property List, expand the "Display and Printing" theme.

3 Select a printing option:

Object Properties window

Property List



Print Frame options

Because there may be more records than can fit in the subform area, 4th Dimension provides the following three check boxes for controlling how records in a subform are printed:

- Variable
- Fixed (truncation)
- Fixed (multiple records)

If you select the **Variable** radio button, 4th Dimension expands or contracts the subform area to print all the records.

If you select the **Fixed (truncation)** radio button, 4th Dimension prints only the records that fit into the area of the subform. The form is printed only once and those records that are not printed are ignored.

If you select the **Fixed (multiple records)** radio button, the frame remains the same size, but 4th Dimension prints the form several times to include all the records.

Note You cannot place any objects on either side of a variable frame subform. Objects placed on either side of the subform will be repeated for every line of the subform.

Printing Picture Fields

Pictures can be printed with either fixed or variable frames.

► To set a print option for a picture field:

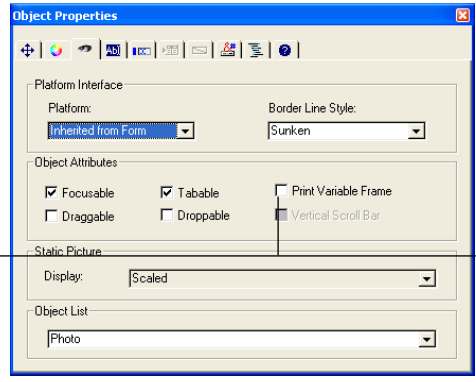
1 **Double-click the picture field in the form.**

Depending on your current display settings, either the Object Properties window or the Property List appears. For more information on this choice, refer to "Displaying and Setting Form and Object Properties" on page 282.

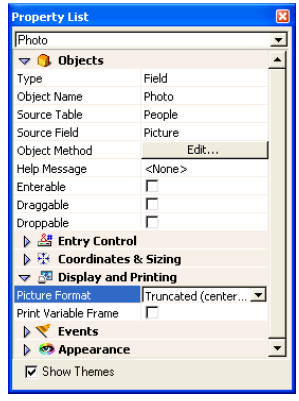
2 **In the Object Properties window, click the Display tab.**
OR
In the Property List, expand the "Display and Printing" theme.

3 **Select a Printing option:**

Object Properties window



Property List



Print Variable Frame Option

4 **Select the Print Variable Frame check box to print the picture with a variable frame.**

When the **Print Variable Frame** check box is selected, the print frames expand to show the entire picture.

OR
Deselect the check box to print the picture with a fixed frame.

Printing Text Fields

Text fields can be printed with either Variable frame or Fixed frame. Print options for Text fields are selected exactly as they are for Picture fields.

▶ To select a print option for a Text field:

1 **Double-click the Text field in the form.**

Depending on your current display settings, either the Object Properties window or the Property List appears. For more information on this choice, refer to "Displaying and Setting Form and Object Properties" on page 282.

2 **Select the Print Variable Frame check box to print the text field with a variable frame.**
OR
Deselect the Print Variable Frame check box to print the text field with a fixed frame.

If the **Print Variable Frame** check box is selected, the Text field expands dynamically to display as many lines as necessary to accommodate the amount of text that was entered.

Printing Labels

You can generate labels with either the Label editor in the User environment or a custom report form. If you use a report form, you have more extensive customization options. In particular, you can use the Form editor to insert variables within your labels.

Once you have created a label report form in the Form editor of 4th Dimension, you can use it in two ways:

- **Using the PRINT LABEL command**
The PRINT LABEL(table) statement causes the printing of the current selection of *table* in the current List form. In this case, 4th Dimension uses the markers (width and height) of this form to set the label format.

- **Using the Label editor**
This operation allows you to benefit from both the advanced functions of the Form editor and the layout parameters of the Label editor.
The Label editor only takes the absolute position of the objects present in the form as well as any margins set into account. The position of the markers is ignored.

Using the PRINT LABEL command

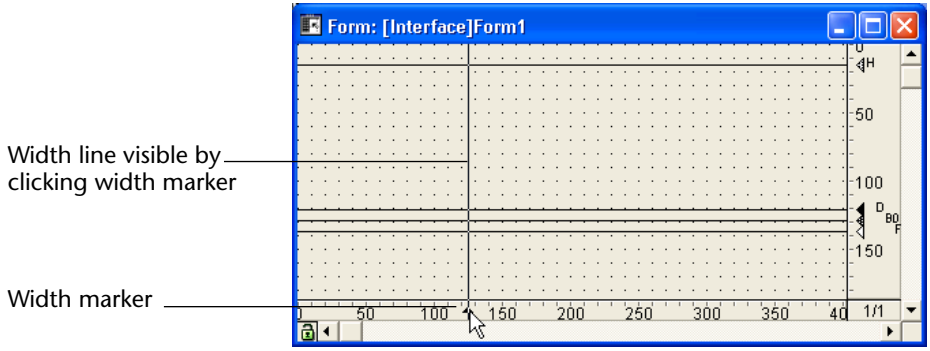
► To create a label report form and print it using the PRINT LABEL command:

1 Set the label width by dragging the width marker on the bottom ruler of the Form editor.

The width marker determines how many labels the PRINT LABEL command prints across the page. The placement of the width marker should correspond to the width of your labels. Be sure to take into account the left and right margins of your labels.

You can determine these margins by calculating the space between each label and dividing this number by two.

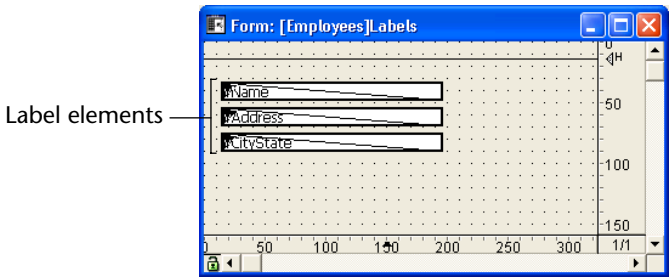
If you have labels that are 2.25 inches wide with a margin of .125 inches (1/8 inch) on either side of each label, the label width marker should be placed at 2.625 inches (2 3/8 inches) to ensure that the label text is placed correctly on the labels.



Note You may want to change the ruler units in the form to inches to make it easier to determine the proper placement of the label width marker. To change the ruler units, choose **Define Ruler Units** from the **Form** menu and select Inches.

2 Insert and set the objects making up each label.

Labels may contain fields and active objects (with which you can associate methods if desired), as well as graphic objects and static text. A subform cannot be printed in a label.



This label contains variables (active objects) whose values are calculated by object methods. For example, the method of the *Full Name* variable concatenates the first and last name of each person and places a space between them:

```
Full name:=[Clients]FirstName+" "+[Clients]LastName
```

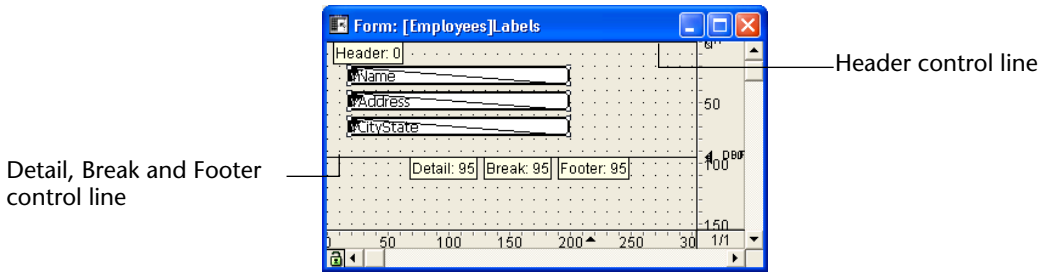
For more information about object methods, refer to chapter 8.

3 Position the objects in the form with respect to the width marker.

The label elements must be placed to the left of the width marker. They must be centered between the left edge of the form and the marker in order for the space on each side of the label to be the same.

4 Set the control lines so that the Header control line is above the label and the Detail, Break, and Footer control lines are below the label.

The Header control line should be set at zero inches and the Detail control line should be set to the height of the label. To center the text within the label, center the form elements between the Header control line and the Detail control line.



When you print the labels, everything between the Header and the Detail control lines appears on the labels.

5 **Save the form and print the selection using the PRINT LABEL command.**

For more information, refer to the *Language Reference* manual of 4th Dimension.

Using the Label editor

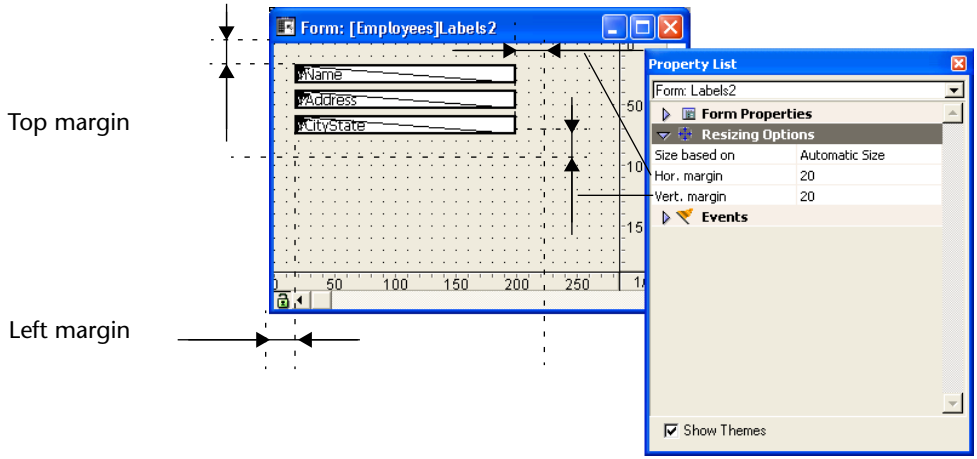
► To create a label report form and print it using the Label editor:

1 **In the form, insert and set the objects making up each label.**

Refer to step 2 in the previous paragraph.

2 **Position each element in the label according to the following principles:**

- the left and top margin of each label is determined by the position of the elements with respect to the (0,0) coordinates of the form.
- the right and bottom margin of each label is determined by the **Hor. margin** and **Vert. margin** values set in the Property List of the form.



3 **Return to the User environment to print the Labels.**

4 **Choose the label form from the List of Tables window to make it the current output form for the table.**

You can display the List of Tables window by pressing **Ctrl+Space bar** (Windows) or **Command–Space bar** (MacOS).

Note If you have multiple operating systems installed on your PC and are using **Ctrl+Space bar** to switch between systems, use **Ctrl+Shift+Space bar** to display the List of Tables window.

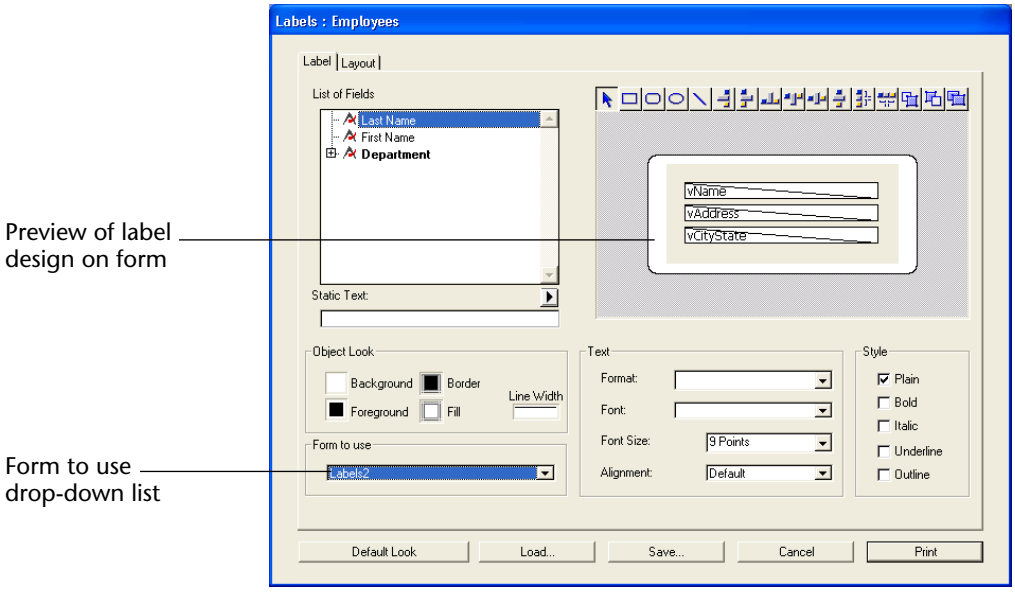
For more information about using the List of Tables window, refer to the *4th Dimension User Reference*.

5 Choose Labels from the Report menu.

The Label Wizard appears.

6 Choose the label form you designed from the "Form To Use" drop-down list.

This tells 4th Dimension to use this form to print the labels:



7 Make any other changes to the Label editor such as setting the label margins or specifying the font in which the text appears.

For complete information about using the Label editor, refer to the *4th Dimension User Reference*.

8 Click the Print button.

4th Dimension prints the current selection of records using your label report design.

8 Creating Methods

You can attach a method to a 4th Dimension object to specify the object's action. A method is a series of instructions that tell the object to do something. For example, you can use methods to:

- Enforce business rules during data entry,
- Calculate values for fields and variables,
- Manage interface elements such as combo boxes, hierarchical lists, and tab controls,
- Manage drag and drop actions,
- Assign actions to custom menu commands,
- Create and manage multiple processes,
- Manage transactions,
- Manage custom reports,
- Regulate multi-user database access.

This chapter provides information about using 4th Dimension's Method editor to create and modify methods. To learn more about 4th Dimension's programming language, refer to the *4th Dimension Language Reference* manual, which provides detailed information about the programming commands and syntax.

4th Dimension Methods

You can create the following four types of methods:

- **Object methods** Object methods are associated with individual objects on a form, such as fields, buttons, drop-down lists, and tab controls. They can be used for such purposes as assigning initial values, managing and validating data entry, or managing drag-and-drop actions.
- **Form methods** Form methods are attached to individual forms. A Form method can manage everything that happens when a form is used for data entry, screen display, or printing. Alternatively, you can use Object methods to manage individual objects on the form.
- **Table methods** Table methods (triggers) are run when specific events occur at the database engine level. For more information, see "Triggers" on page 519 and "Triggers" on page 524.
- **Database methods** Database methods run automatically when certain worksession-related events occur. For more information about Database methods, see "Database Methods" on page 521.
- **Project methods** Project methods can be called by other methods anywhere in the database, by users, by the Web server or by custom menu items.

Object Methods

An object method is attached to a field or other active object on a form. You create object methods in the Form editor. The object method executes when certain events occur. The method is bound to the object to which it is attached and moves with it when copied.

An object method can perform calculations, get related information from other tables, concatenate data, validate data, display a specific page in a multi-page form, and more. The following are some examples of things you can do in an object method:

- Display the current date on screen or print it in a report,
- Manage data entry from a combo box,
- Display a custom dialog box when the user presses a button on a form,
- Specify the action when a user drags an item in a scrollable list to another object.

An object method can perform data entry control similar to the built-in data entry features (in the Data Control section of the Properties Window). For example, you can write an object method to perform data validation, set the display format, or establish an entry filter.

Form Methods

A form method is a method attached to a form. Each form can have one form method. It is executed whenever the form is used. The method is bound to the form, but does not move with it when a form is copied (form elements are copied, not entire forms).

Note Only certain events occur when a form is used as an output form. For information about form events, refer to the *4th Dimension Language Reference* manual.

A form method manages a form at a higher level than do object methods. Form methods are used to control the interaction between different objects and the form as a whole. One use for a form method is to calculate values based on more than one field. Since the calculation must be performed whenever any of the values involved changes, you place the calculation in a form method.

For example, you might place the following statement in a form method:

```
vTax:= TotalSales * TaxRate
```

As a form method, this statement can be executed every time anything on the form changes. This ensures that the variable vTax is always up-to-date.

Triggers

Triggers are methods that run automatically when certain events occur at the database engine level. Those events are:

- **On Saving New Record** The trigger will be invoked when a record is added to the table. This happens when:
  - Adding a record in data entry (in the User environment or using the ADD RECORD command).
  - Creating and saving a record with CREATE RECORD and SAVE RECORD. The trigger is invoked when you call SAVE RECORD, not when it is created.
  - Importing records (in the User environment or using an import command).

- Calling any other commands that create or save new records (i.e., ARRAY TO SELECTION, SAVE RELATED ONE, etc.)
- Using a plug-in that calls the CREATE RECORD and SAVE RECORD commands.

- **On Saving an Existing Record** The trigger will be invoked when a record of the table is modified. This happens when:
  - Modifying a record in data entry (in the User environment or MODIFY RECORD command).
  - Saving an already exiting record with SAVE RECORD.
  - Calling any other commands that save existing records (i.e., ARRAY TO SELECTION, APPLY TO SELECTION, MODIFY SELECTION, etc.).
  - Using a plug-in that calls the SAVE RECORD command.
- **On Deleting a Record** The trigger will be invoked when a record of the table is deleted. This happens when:
  - Deleting a record (in the User environment or calling DELETE RECORD or DELETE SELECTION).
  - Performing any operation that deletes related records through the Deletion control options of a relation.
  - Using a plug-in that calls the DELETE RECORD command.
- **On Loading a Record** The trigger will be invoked when a record of the table is loaded. This includes all situations in which a current record is loaded from the data file[1]. You will this option less often than the three previous ones.

Note This option does not cover the creation of a new record; it only applies to the loading of existing records.

Note to users of former versions of 4D

The trigger is a new type of method introduced in version 6. In previous releases of 4th Dimension, table methods (called File procedures) were executed only when a form for a table was used for data entry, display, or printing — they were rarely used. Triggers execute at a much lower level than File procedures. No matter how a

1. This trigger is not invoked when using functions that may use indexes. If the index is used, records are not loaded and, conversely, if the index is not used, records are loaded. This uncertainty about the loading of the records does not allow the trigger to be managed properly. It is therefore never invoked in this case. For more information, refer to the *4th Dimension Language Reference* manual.

record is created, modified, or deleted — by user actions (such as data entry) or programmatically (such as a call to SAVE RECORD) — the trigger will be invoked.

Database Methods

Database methods run when certain worksession-related events occur. Those events are:

- On Startup (equivalent to the *STARTUP* procedure in previous releases of 4th Dimension)
- On Web connection
- On Server startup
- On Server shutdown
- On Server open connection
- On Server close connection
- On Exit

4th Dimension ships with empty methods that are associated with these events. You can add code to any or all of these methods. For complete information on the uses of database methods, see the section "Database Methods" in the *4th Dimension Language Reference* manual. The specific uses of database methods with the Web Server are described in the "Web Server" section of the *Language Reference* manual. For more information on the specific uses of database methods with 4D Server, refer to the *4D Server Reference Manual*.

Note If you are converting a database written with an earlier version of 4th Dimension, it may have a *STARTUP* procedure. If you want that procedure to run automatically at startup, select "Use V3.x.x Startup Method Scheme" in the **Options** page ("Design mode" theme) of the Preferences dialog box and ignore the new database methods architecture[1]. Alternatively, you can deselect this option, paste the code from your *STARTUP* procedure into the On Startup database method, and use the new database methods scheme.

1. For more information on this option, see the section "Options Page" on page 127.

Project methods

A project method can be called by any other method, associated with a menu item or executed by the Web server via tags or special URLs. A project method can also be executed by choosing **Execute Method** from the User environment's **Special** menu. When a project method is called by another method, the statements in the project method are substituted for the method's name in the method which is calling it.

Project methods are not associated with a table, form, or object. You can create as many project methods as you need.

The following is an example of a project method:

```
  `Adds a record to the [Employees] table
INPUT FORM ([Employees];"Input")
OK:=1
While (OK=1)
  ADD RECORD ([Employees])
End while
```

This method is attached to a menu item that is used for entering new records. The method lets the user add new records to the [Employees] table using the form named Input. The user is able to add new records until he or she clicks the Cancel button[1].

A method can also be used as a formula that you apply to the current selection. For information about applying formulas, refer to the *4th Dimension User Reference*.

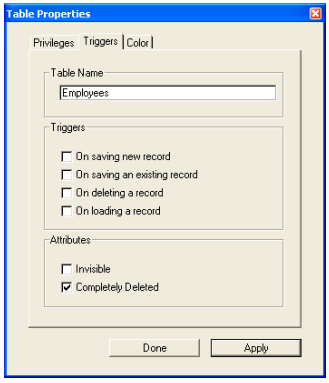
Events

Database, Table (trigger), Form, and Object methods run only when certain events occur. You specify those events when you set the properties for each object. The Table, Form, and Object Properties windows each have pages in which you specify the events that will run for the object.

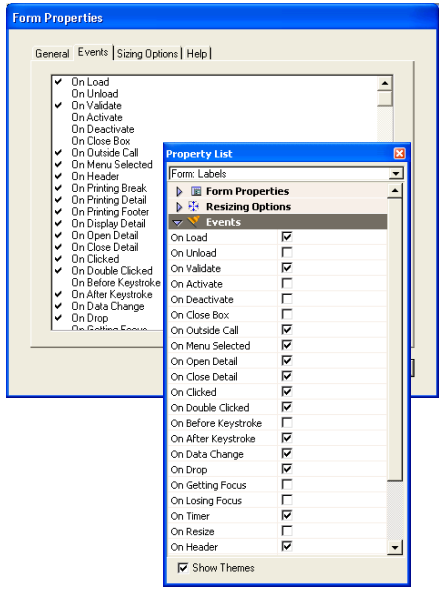
1. Clicking Cancel sets the system OK variable to 0.

The following illustrations show those pages:

Triggers page of Table properties window

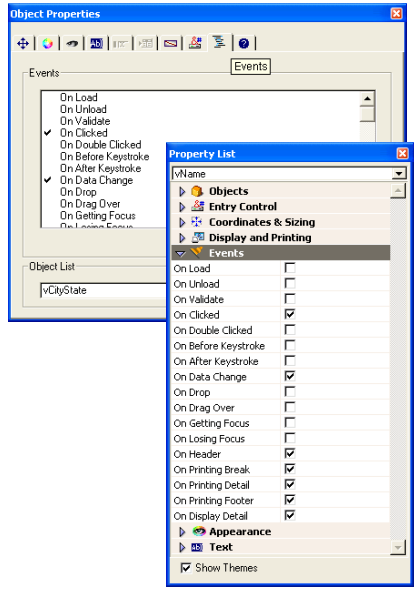


Events page of Form properties window



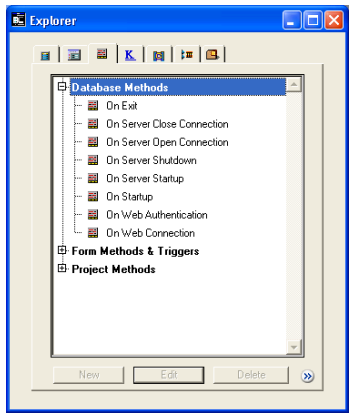
Events theme of property List (forms)

Events page of Object properties window



Events theme of property List (object)

In addition, 4th Dimension provides blank Database methods which are listed in the Methods page of the Explorer. Each Database method runs only when the corresponding worksession-related event occurs. You can open a blank database method from the Methods page of the Explorer:



Database Methods

Database methods are executed when events corresponding to their names occur:

- **On Server Shutdown**: the 4D Server application is about to quit.
- **On Web Authentication**: reception of an HTTP request.
- **On Web Connection**: the HTTP request has been accepted.
- **On Server Startup**: the 4D Server application is in the process of starting up.
- **On Exit**: the database is about to quit.
- **On Server Close Connection**: a client machine disconnects from 4D Server.
- **On Startup**: the database is in the process of starting up.
- **On Server Open Connection**: a client machine connects to 4D Server.

You write a Database method by opening the desired blank Database method and entering your code.

Triggers

Table events (triggers) are run when particular events regarding saving, deleting, and loading records occur. If you want a trigger to run when a particular event occurs, you must check the appropriate event in the Triggers page of the Table properties window. You then create the trigger by creating a Table method from the Explorer. For more information about creating a Table method, see the section "Creating a Trigger" on page 537.

In the Trigger, you generally need to test for each event that you checked in the Triggers page. To do so, you can use a Case statement and the Database event function. The Constants page of the Explorer lists constants associated with each Table event. The Constants page of the Explorer window lists the constants associated with each trigger in the Database Events theme.

Your Case statement shell would look like this:

```
Case of
  :(Database event=On Saving New Record Event)
     ` Perform actions for saving a new record
  :(Database event=On Saving Existing Record Event)
     ` Perform actions for saving an existing record
  :(Database event=On Deleting Record Event)
```

```
      ` Perform actions for deleting an existing record
   :(Database event=On Loading Record Event)
      ` Perform actions for loading a record into memory
End case
```

Note The **Options** page ("Design mode" theme) of the Preferences dialog box has an option that lets you run Table methods (triggers) according to the rules that were established for File procedures in former versions of 4th Dimension. If you wish to use Table methods in this manner, select the Use V3.x.x Startup Method Scheme.

A trigger has two basic functions:

- Perform actions on a record before it is saved, modified, deleted, or just after it is loaded.
- Accept or reject a database operation, such as a save record operation.

For complete information on using triggers, see the *4th Dimension Language Reference* manual.

Form and Object Events

Form and object methods run when particular form and object-level events occur. If you want a form or object method to run when a particular event occurs, you must activate the appropriate event in the Events page of the Form or Object Properties window.

The following is a list of events for forms used for screen display:

- **On Load** 4th Dimension is about to display the form on-screen or print the form.
- **On Unload** The form is about to be closed and released.
- **On Validate** After the user has accepted the record.
- **On Clicked** The user clicks the object that has focus.
- **On Outside Call** When the form receives a call from CALL PROCESS.
- **On Activate** When the form's window becomes the frontmost window.
- **On Deactivate** When the form's window is no longer the frontmost window (i.e., when another window becomes the frontmost window).
- **On Double-clicked** The user double-clicks the object.

- **On Getting Focus** When a form object gets focus (i.e., the user presses Tab to select the object or clicks the object).
- **On Losing Focus** When a form object loses focus (i.e., the user presses Tab to select the next object in the entry order or clicks another object to select it).
- **On Drop** When the object receives a dragged object.
- **On Drag Over** When an object is dragged over the object.
- **On Before Keystroke** When the user presses a key in the object. The Get edited text function returns the contents of the object that has the focus.
- **On After Keystroke** When the user presses a key in the object. The Get edited text function returns the contents of the object that has the focus, including the last keystroke.
- **On Menu Selected** A menu item has been chosen.
- **On Plug-in Area** A plug-in requested that its object method be executed.
- **On Data Change** When the data in the object changes.
- **On Close Box** The window's Close box has been clicked.
- **On Display Detail**[1] A record is about to be displayed in an output form.
- **On Open Detail**[1] A record is double-clicked in an output form and 4th Dimension is about to display the record in the current input form.
- **On Close Detail**[1] 4th Dimension is putting away the input form and is about to redisplay the output form.
- **On Timer** The number of ticks set by the SET TIMER command was reached.
- **On Resize** When the form's window is resized.
- **On Header** When the form's header is either being displayed or printed.

For printed reports only, the following events occur:

1. These events operate only when using the MODIFY SELECTION and DISPLAY SELECTION commands.

- **On Printing Detail** The form's Detail area is about to be printed.
- **On Printing Break** The form's Break areas are about to be printed.
- **On Printing Footer** The form's Footer area is about to be printed.

For complete information on form events, refer to the discussion of the Form event function and the form event constants in the *4th Dimension Language Reference* manual.

When you write a form or object method, you generally need to test for each event that you activated in the Form or Object Properties window. To do so, you can use a Case statement and the Form event function. The Constants page of the Explorer lists constants associated with each Form event. Your Case statement would look like this:

```
Case of
  :(Form event=On Load)
     ` Perform appropriate actions here...
  :(Form event=On Data Change)
     ` Perform appropriate actions here...
  :(Form event =On Validate)
     ` Perform appropriate actions here...
  :(Form event=On Clicked)
        ` Perform appropriate actions here...
.
.
.
End case
```

Introduction to Methods

A 4th Dimension method is a series of instructions that causes 4th Dimension to perform an action or a series of actions. For example, the following project method allows users to add records to a table:

```
INPUT FORM ([Customers];"Input")
Repeat
  ADD RECORD ([Customers])
Until (OK=0)
```

This method would be attached to a menu item in a custom application. When the user chooses that menu item, 4th Dimension runs this method. It makes the [Customers]Input form the current Input form and presents it to the user as a blank form, ready for data entry.

The user could continue to enter new record until he or she presses the Cancel button on the form. During data entry, any table, form, or object methods would run when the appropriate events occur.

4th Dimension methods are created with components of a *procedural language*. The following are the main elements of the language:

- **Fields** You can use fields from any table. For example, a method can use a value that is stored in a field or it can change that value and store a new value in that field.
- **Object names** You can use the names of objects on a form. For example, you can resize an object, change its color, enable or disable buttons, or modify the font, font size, or style.
- **Variables** You can temporarily store a value in a variable and use it later in the same method or in a different method. You can create a variable in any method and you can use or change its value in any method.
- **Pointers** Pointers let you write generic code that doesn't refer to database objects by name. Instead, a pointer to each object is used. Each time the generic code is used, you can "point" to different database objects. For example, if you substitute a pointer to a table for the table name "[Customers]" in the previous example, you could then reuse the code for any table. For more information on pointers, see the chapter "Arrays and Pointers" in the *4th Dimension Language Reference* manual.
- **Operators** You can use symbols to instruct 4th Dimension to carry out an operation such as multiplication, addition, and so on.
- **Commands** You can use commands in the language to instruct 4th Dimension to perform an action. For example, the ALERT command displays a message in an alert dialog box. The NEXT PAGE command displays the next page of a multi-page form and the ORDER BY command sorts the records in the current selection.
- **Functions** You can use functions in the language to calculate values. For example, you can calculate an average of several values with the Average function. You can calculate a subtotal for a report with the Subtotal function.

- **Flow of control** You can control when code executes with flow of control structures. The 4th Dimension language includes the following control structures:
  - If…Else…End If
  - Case of…Else…End Case
  - While…End While
  - Repeat…Until
  - For…End For

  You use logical tests in these structures to determine whether or how many times code executes. All of these elements are discussed in detail in the *4th Dimension Language Reference* manual.

Examples

This section describes features common to all types of methods.

Statements

A method is composed of statements, each statement consisting of one line in the method. A statement is an instruction for 4th Dimension to carry out. For example, the following line is a statement:

[People]Start Day:= **Current date**

This statement places the current date in the Start Day field of the [People] table. Current date is a function that returns the date based on the system date. [People]Start Day is a field.

Notice that the statement specifies the table name, surrounded by square brackets, with the field. When writing project methods, you specify the table name to which a field belongs to avoid possible confusion with other fields with the same name. However, when in a form or object method, you can specify fields from the form's table without specifying the table name. Table names are written within square brackets.

The previous statement is typical of statements that calculate or work with values. It starts with the field in which the value is to be placed and uses the assignment operator to point at the calculation that determines the value. The calculation is performed by whatever follows the assignment operator. The assignment operator is a colon and equals sign (:=).

You use the assignment operator whenever you need to store a value in a field, an object, or a variable. It takes the following form:

Field/Object/Variable := *Calculation*

The value container is the field, object, or variable in which you want to place the value. The calculation is the operation that results in the value you want to store. The assignment operator assigns the value that is calculated on its right into the container on its left. You will see several examples of this in the next few paragraphs.

A statement may be simple or complex. Although a statement is always one line, that one line can be as long as needed (up to 32,000 characters).

The following method displays the third page in a multi-page form:

GOTO PAGE (3)

When you use the GOTO PAGE command, you instruct 4th Dimension to display the page indicated in the parentheses.

Notice that the command is in bold capital letters; this is the way that 4th Dimension displays commands in the Method editor by default. This convention is used in all examples in the 4th Dimension documentation. You can change the style of each type of object in the application Preferences.
You do not have to type commands in all capital letters; 4th Dimension automatically changes the display.

Most commands require additional information to carry out the instruction. This additional information is called an argument to the command. An argument contains data that a command needs in order to complete its task. In this case, the GOTO PAGE command needs the page number to go to. An argument is always placed within parentheses following a command.

Executing the Method

The execution of statements in a method always follows a strict order. When the method is executed, it begins at the first line and works its way down to the last line (Obviously, the order of execution is determined by the flow of control structures in the method).

This section examines a multi-line method in detail in order to establish some of the terminology, concepts, and common aspects of methods.

The following method computes the total amount due on an invoice:

```
vSales Tax:= Total Purchases * Tax Rate
vTotal:= Total Purchases + vSales Tax
[Report]Total Due:= Round (Total; 2)
```

This method is attached to an object that will be printed on an invoice. As you follow the method line-by-line, you will see that the later lines depend on previous lines. Because of the strict order in which statements are executed, you can depend on a value being available when it is needed.

Here is the first line of the method. It calculates the sales tax for the purchase:

```
vSales Tax:= Total Purchases * Tax Rate
```

In this method, "vSales Tax" is a variable. You can create a variable at any time by typing its name. You name the variable on the left of the assignment operator and calculate a value on the right. Subsequently, whenever you need that value, you can simply use the variable name. A variable can be created at any time in a method. It can be assigned a value, as here, which can then be used by a later statement.

The variable "vSales Tax" is a *process variable*. A process variable is a variable which works within the current process. In 4th Dimension, everything occurs within a process, even if the process is one created by 4th Dimension. (You will learn about processes in chapter 13.).

When a statement creates a process variable such as vSales Tax, a portion of memory is set aside and assigned the name "vSalesTax." Since it is a process variable, any method in the current process can use the value in vSalesTax. The variable remains in memory until the process is closed. *Local* variables temporarily store values that can be used within the method, but which cannot be used by any other method.

Variables follow the same rules of naming as fields. Typically, you would use a consistent convention so that you know that you are using a variable. In the previous example, the variable is indicated by a lowercase v as the first letter in the variable vSales Tax. For more information about variables and the different types of variables, refer to the *4th Dimension Language Reference* manual.

For the vSales Tax variable the assignment operator (:=) assigns it the result of a calculation. In this case we would say, "vSales Tax *gets* Total Purchases times Tax Rate." The Total Purchases field contains the total amount of purchases for the current invoice and the Tax Rate field contains the tax rate to be used to calculate the tax. (A Tax Rate field could contain different rates based on the address of the customer.)

When an object method uses field names, the instruction is to use the value in that field on the current record. When another record is displayed, used, or printed, the method is executed again, using values in the new record.

The first line of our example method multiplies the values in the Tax Rate and Total Purchases fields and stores the result in the variable vSales Tax for each record that is used (a value entered or an invoice printed).

```
vSales Tax:= Total Purchases * Tax Rate
```

Here is the second line of the method. It creates a variable that contains the grand total of purchases plus sales tax:

```
vTotal:= Total Purchases + vSales Tax
```

The first component of the statement above is the vTotal variable. It is assigned the value generated by the addition of Total Purchases (a field) and vSales Tax (a variable). It does not matter to 4th Dimension whether a value is stored in a field or in a variable. As long as the field and the variable store data of the same type, the addition operator (+) simply adds the two values together.

Notice that this statement uses the vSalesTax variable that was created in the previous line. It also creates the vTotal variable which will be used in the following line.

Notice also that the value of Total Purchases has been used twice: first in the previous line to calculate the vSales Tax, and second in this line to calculate the vTotal. Nothing happened to the value in the field in either case. In each of these two lines, the assignment operator placed a value in a variable. These statements use the value in the Total Purchases field, but do not change that value.

The assignment operator (:=) places a value in the field, object, or variable to its left. Nothing will happen to fields, objects, or variables to the right of the assignment operator.

Here is the third line of the method. It stores a value in a field in a different table:

[Report]Total Due:= **Round** (vTotal; 2)

The first component is a field. Notice that the table name has been specified because it is different from that of the master table. Table names are always placed in square brackets [like this]. If you have to use parentheses as well, you would use both ([like this]).

The calculation is performed by the Round function. The Round function rounds off a value to a specified number of places.

Most functions require additional information to carry out the calculation. Additional information for a function is the argument to the function. A function may have several arguments. In this case, the Round function needs two arguments: the number to round and another number specifying how many places to round to. Here, the number to round is the value of the vTotal variable, and the number of places to round to is 2. The two arguments are separated by a semicolon. The variable vTotal and the number 2 are *arguments* to the Round function.

Notice that the function's name has only the first letter capitalized. This is the naming convention used for 4th Dimension functions. Just as with commands, you do not need to enter the capital letter. When you press the **Enter** key or click on another line in the method, 4th Dimension automatically capitalizes the first letter of the function and makes it bold (a convention which can be modified in the Preferences).

Where to Put an Object Method

You can attach an object method to any field or other active object. An object method can refer to values in other fields and objects. The general rule is to attach an object method to the active field or object, the one that receives the data entry or the one that is clicked or otherwise activated. An object method that is to be executed when a button is clicked should be attached to the button object. An object method that capitalizes entries in a field should be attached to the field.

However, if you placed the method

Grand Total:= Total + Sales Tax

in the Grand Total field, the method would not function properly since you would have to type something in the Grand Total field in order for the method to be executed.

You need the statement to be executed whenever the values in the Total or Sales Tax fields change. In order for the calculation to take place, you should place the statement in the form method or use it in object methods for both the Total and Sales Tax fields.

Creating or Opening Methods

Methods are created using the Method editor. The Method editor provides you with tools to create, test, and edit any type of method.

4D Server Object locking occurs when two or more users attempt to modify the same method at the same time. If a user opens a method in the Design environment, the method is locked. Other users cannot modify this same method until the first user frees it by closing the window. In the meantime, the method can be opened in read-only in order to copy all or part of its elements.

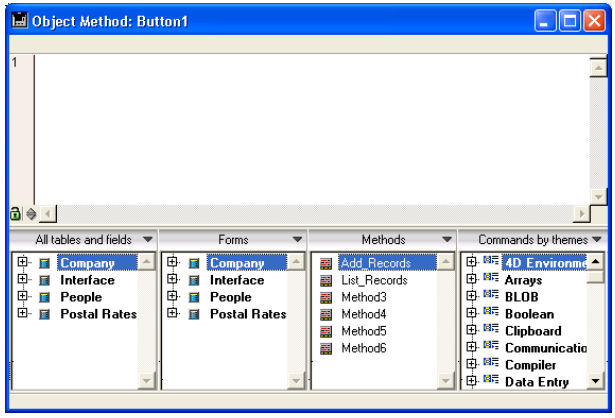
Creating an Object Method

Object methods are created for an object on a form. You start in the Form editor, with a form displayed on the screen.

► To create or open an object method in the Form editor:

1 **In the Form editor, hold down the Alt key (Windows) or the Option key (MacOS) and click the object.**
OR
Select the object, then choose Object Method from the Object menu.
OR
Click the object using the right mouse button (Windows) or press the Control key while clicking the object (MacOS) and choose Object Method from the contextual menu.
OR
In the Object Properties window or the Property List, click the object method Edit... button.

A window of the Method editor appears, blank if you have just created the method.



The Method editor is described in the section "Using the Method Editor" on page 548.

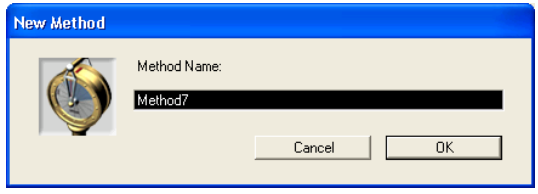
Creating or Opening a Project Method

You can create or open a project method using the **Design** menu, the Methods page of the Explorer, the Menu Bar editor or a window of the Method editor.

▶ To create a project method from the **Design** menu:

1 **Choose New Method from the Design menu.**

4th Dimension displays the New Method dialog box.



2 **Enter a method name.**

Method names can be up to 31 characters long. They can include letters, numbers, the space character, and the underline character.

3 **Click the OK button.**

4th Dimension opens an blank Method editor window where you can begin writing the new method.

► To create a project method from the Explorer:

1 **Click the Methods tab in the Explorer.**
The hierarchical list of Database, Form/Triggers and Project methods appears.

2 **Highlight the Project Methods item.**

3 **Click New.**
The New Method dialog box appears.

4 **Enter the method name and click OK.**

► To open a project method from the **Design** menu:

1 **Choose Edit Method... from the Design menu.**
The Methods page of the Explorer appears (see below).

► To open a project method from the Explorer:

1 **Click the Methods tab in the Explorer.**

2 **Expand the Project Methods element, then select the name of the method to be opened.**

3 **Click the Edit button.**
OR
Double-click the name of the method to be opened.
The method opens in a window of the editor.

► To create or open a project method from the Menu Bar editor:

1 **In the Menu Bar editor, click in the "Method Name" area for an item of the current menu bar.**
This area contains the name of the method associated with the selected item (if any).

2 **Type Ctrl+P (Windows) or Command+P (MacOS).**
- If the project method does not exist, the Methods page of the Explorer is displayed.
- If the project method exists, it is displayed in a new window of the Method editor.

► To create or open a project method from a window of the Method editor:

1 **In the Method editor, select the name of the project method to be opened.**

2 **Type Ctrl+P (Windows) or Command+P (MacOS).**

- If the project method does not exist, the Methods page of the Explorer is displayed.
- If the project method exists, it is displayed in a new window of the Method editor.

The Method editor is described in the section "Using the Method Editor" on page 548.

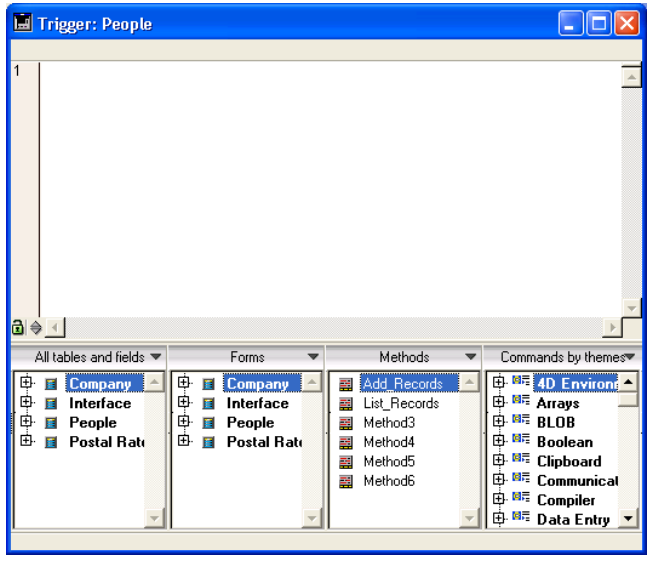
Creating a Trigger

You can create a trigger using a shortcut in the Structure editor or you can create the trigger directly in the Explorer:

► To create or open a trigger from the Structure editor window:

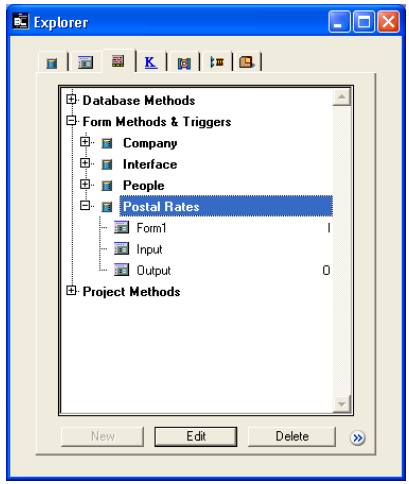
1 **Hold down the Alt key (Windows) or Option key (MacOS) and double-click the table title in the Structure editor window.**

A Method editor window appears, blank if you have just created the trigger:



► To create or open a trigger from the Explorer:

1 **Click the Methods tab in the Explorer window and expand the "Form Methods & Triggers" element.**



2 **Highlight the desired table and click Edit.**
OR
Double-click on the desired table.
The trigger appears in a window of the Method editor.

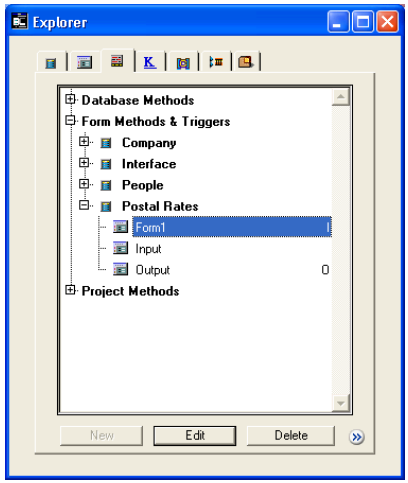
Creating or Opening a Form Method

Form methods can be opened from the Methods page of the Explorer or from the Form editor.

► To create or open a Form method from the Explorer:

1 **Click the Methods tab in the Explorer.**

2 **Expand the table to which the form belongs and highlight the desired form.**



3 **Click Edit.**
OR
Double-click the form.
The form method appears in a window of the Method editor.

Note You can also create or open a form method from the **Forms** page of the Explorer: right-click (under Windows) or Control+click (under MacOS) on the name of the form and choose **Edit Form Method...** in the contextual menu.

► To create or open a form method from the Form editor:

1 **Choose Form Method from the Form menu.**
OR
Click an empty area on the form using the right mouse button (Windows) or press the Control key while clicking an empty area on the form (MacOS) and choose Form Method from the contextual menu.
OR
In the Property List, click the Edit... button located next to the Form Method line.
The form method appears in a window of the Method editor.

Deleting methods

You can delete a trigger, form method or project method at any time using the Explorer.
You can also delete an object method using the Form editor.

On the other hand, you cannot delete database methods. To disable this type of method, erase all the statements in the method or change them to comments by preceding them with the "`" symbol (used to distinguish comments from executable code). For more information about comments in methods, refer to the paragraph "Comment/Uncomment," page 571.

Deleting a Project Method, a Form Method or a Trigger

► To delete a form method, a project method, or a trigger using the Explorer:

1 **In the Explorer click the Methods tab.**

Make sure the Methods page is displayed, since deletion operations cannot be undone.

2 **In the Explorer window, expand the method type that corresponds to the method you want to delete (Form Methods and Triggers or Project Methods).**

3 **Select the table or form to which the method is assigned.**
OR
Highlight the name of the project method you want to delete.

4th Dimension displays the method in the Method editor.

4 **Click the Delete button located at the bottom of the Explorer window.**

Note To delete a project method, you can also **right-click** (under Windows) or **Control+click** (under MacOS) on the name of the project method and choose **Delete Method** from the contextual menu.

4th Dimension displays an alert asking you to confirm the deletion.

5 **Make sure it is the right object you are about to delete and click OK to confirm.**

4th Dimension deletes the corresponding method.

Clearing Unwanted Object Methods

You can clear unwanted object methods using the Form editor. In some cases, clearing unneeded object methods can make the database run faster.

▶ To delete an unwanted object method:

1 **Display the form that contains the method(s) you want to clear.**

2 **Select the object(s) that have unwanted Object methods attached to them.**

3 **Choose Clear Object Method from the Object menu.**
OR
Click on the Clear Object Method icon  **in the Tools palette.**
4th Dimension removes the Object methods from the selected objects.

Note If you clear an Object method by mistake, choose **Undo** from the **Edit** menu.

Defining the properties of project methods

After creating a project method, you can rename it and modify its properties. Project method properties mainly concern their access and security conditions: user access as well as access by the Web server and the Web services.

Note The other types of methods do not have specific properties. Their properties are related to those of the objects to which they are attached.

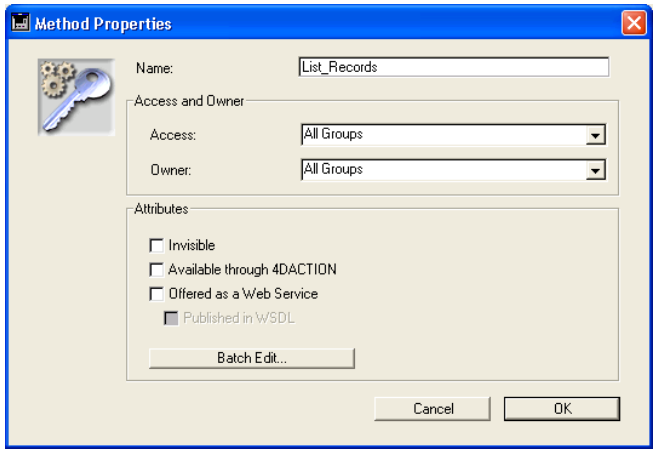
▶ To modify the properties of a project method:

1 **In the Method editor, open the method whose properties you want to set.**

2 **Select Method Properties... from the Method menu.**
OR
Right-click (under Windows) or Control+click (under MacOS) and choose the Method Properties... command from the contextual menu.

Note You can also access the Method properties from the **Methods** page of the Explorer: **right-click** (under Windows) or **Control+click** (under MacOS) on the name of the project method and select **Method Properties...** in the contextual menu.

The Method properties dialog box appears.



3 **Modify the desired properties.**

These properties are detailed in the following paragraphs. The **Batch Edit...** button opens a second dialog box used to modify a property for all or part of the database project methods in a single operation. This dialog box is described in detail in the paragraph "Batch Setting for Method Attributes," page 545.

4 **Click OK to validate the modifications.**

Renaming a Project Method

You can change the name of a project method in the Explorer. Database methods cannot be renamed. The same goes for triggers, form methods, and object methods, which are bound to objects and take their names from the object concerned.

► To rename a project method:

1 **Display the Method properties dialog box.**

2 **Modify the method name and click on OK.**

There is also a shortcut that lets you rename a method using the Explorer.

► To rename a project method using the Explorer:

1 **Display the Methods page of the Explorer.**

2 **Expand the list of project methods so that the one you want is displayed.**

3 **Hold down the Command key (MacOS) or Ctrl key (Windows) and click the project method name.**

The method name becomes editable.

4 **Type a new name.**

5 **Press Tab or click anywhere outside the entry area to save your changes.**

If a method with the same name already exists, 4th Dimension displays a message saying that the method name has already been used. Otherwise, 4th Dimension changes the name of the method and, if necessary, resorts the list of methods.

Note Changing a method name can be done in the Method Properties dialog box. This dialog is described in the following section. Changing a method name can invalidate any methods or formulas that use the old method name. Each such item has to be updated in order to work.

4D Server The method name is changed on the server when you finish editing the name. If more than one user is modifying the method name at the same time, the final method name will be the name specified by the last user to finish editing the name. You may want to specify a method owner so that only certain users can change the method's name.

Access and Owner privileges

You can control access to methods by setting Access and Owner privileges for groups of users. A single group can be assigned for each privilege using the Access and Owner drop-down list in the Method Properties dialog box. For information about creating a password access system with users and groups, see the section "An Access Hierarchy Scheme" on page 617.

The Access drop-down list controls which group can execute the method in the User or Custom Menus environment. If a user that is not in this group attempts to execute the method, 4th Dimension displays a message saying that the user's password does not allow him or her to execute the method.

The Owner drop-down list controls which group can edit the method in the Design environment. If a user who is not in this group attempts to edit the method in the Design environment, 4th Dimension displays a message saying that the user does not have the access privilege to edit the method.

Users who are assigned to both groups can use the form in the User, Custom Menus, and Design environments.

Making a Method Invisible

There are two menu commands in the User environment that give users the ability to run project methods — the **Execute Method** command in the **Special** menu and the **Apply Formula** command in the **Enter** menu. If you don't want any users to run a project method, you can make it Invisible in the Method Properties dialog box. An invisible method does not appear in either the Execute Method dialog box or the Formula editor.

When you make a project method invisible, it is still available to database programmers. It remains listed in the Methods page of the Explorer and in the list of routines in the Method editor.

Note 4th Dimension lets you modify this option for a large number of methods using the **Batch setting of attributes...** option. For more information, refer to the paragraph "Batch Setting for Method Attributes," page 545.

Available through 4DACTION

This option can be used to reinforce 4D Web server security: when it is not checked, the project method concerned cannot be executed via an HTTP request containing special URLs used for calling 4th Dimension methods (4DACTION and 4DMETHOD). For more information about this, refer to the *Language Reference* manual of 4th Dimension.

The project methods available using 4DACTION are given a specific icon:  .

This option is unchecked by default for databases created with 4th Dimension starting with version 2003. Methods that can be executed using the 4DACTION or 4DMETHOD Web URLs must be indicated individually. Conversely, for reasons of compatibility, this option is checked for existing databases (created with a version of 4D earlier than 2003): by default, all the project methods are accessible via the special 4D URLs.

Note 4th Dimension lets you modify this option for a large number of methods using the **Batch setting of attributes...** option. For more information, refer to the paragraph "Batch Setting for Method Attributes," page 545.

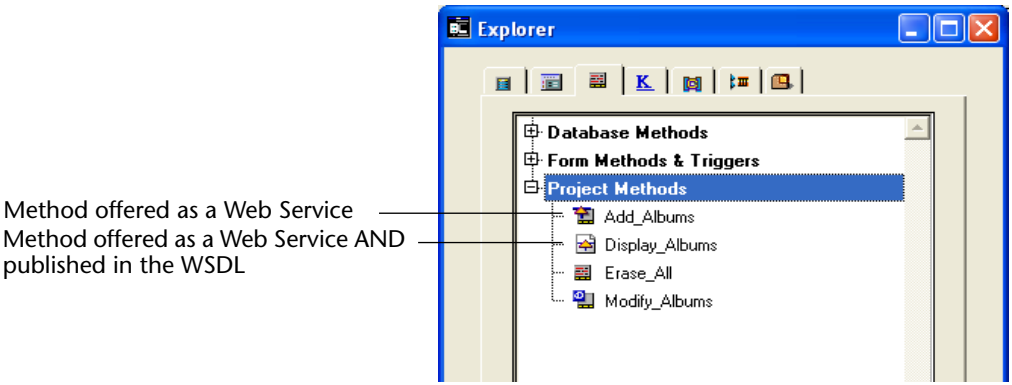
Offered as a Web Service

This attribute lets you publish the current method as a Web Service via a SOAP request. For more information, refer to the paragraph "Publishing a Web Service with 4th Dimension," page 728.

When this option is checked, the **Published in WSDL** option is enabled.

Published in WSDL

This attribute lets you include the current method in the WSDL of the 4th Dimension application. For more information about this, refer to the paragraph "Generation of the WSDL," page 731.



Note 4th Dimension lets you modify this option for a large number of methods using the **Batch setting of attributes...** option. For more information, see the following paragraph.

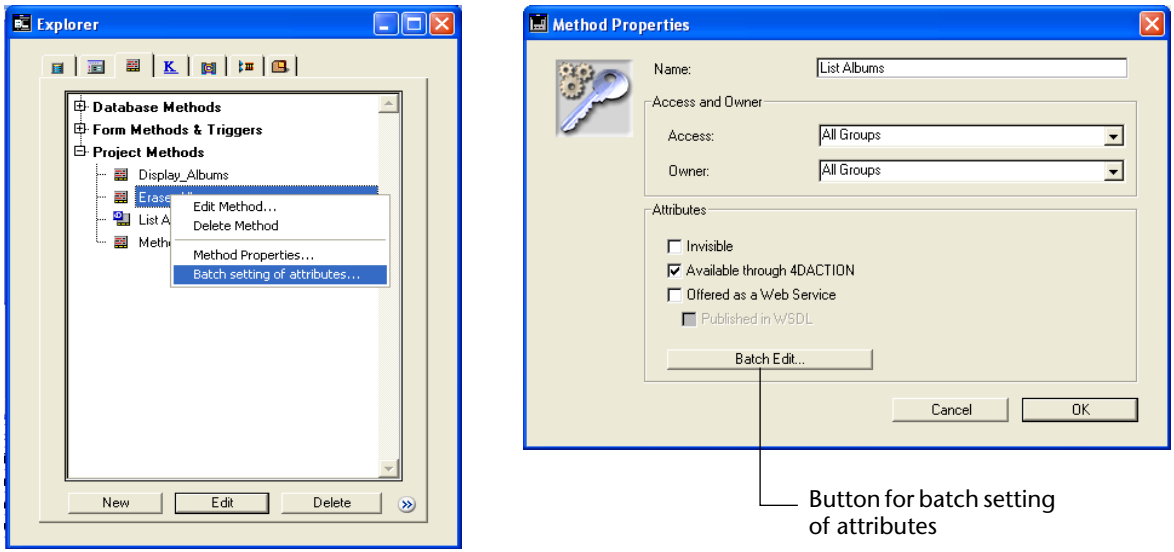
Batch Setting for Method Attributes

4th Dimension offers a new dialog box that is used to modify an attribute (Invisible, Available through 4DACTION, etc.) for all or part of the database project methods in a single operation.

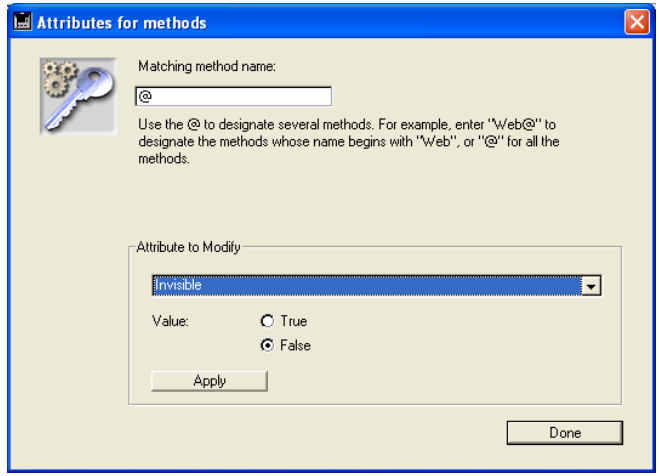
This new feature is especially useful for modifying project method attributes in databases converted from previous versions of 4D. It can also be used during development to apply common attributes to groups of similar methods quickly.

► For batch setting of method attributes:

1 **In the 4D Explorer, click with the right button (Windows) or Ctrl+click (MacOS) on a project method, then choose the Batch setting of attributes... command**
OR
Click on the Batch Edit... button in the Method Properties window:



The "Attributes for methods" dialog box appears:



2 **In the "Matching method name:" area, enter a string that lets you designate the methods you want to modify as a batch.**
The character string is used as a search criterion for the method names.

Use the wildcard character @ to help define groups of methods:

- to designate methods *whose names begin with...,* type @ at the end of the string. For example: web@
- to designate methods *whose names contain...,* type @ in the middle of the string. For example: web@write
- to designate methods *whose names end with...,* type @ at the beginning of the string. For example: @write
- to designate all of the methods, just type @ in the area.

Notes • The search does not take upper/lower case into account.
• You can enter several @ characters in the string, for example dtro_@web@pro@

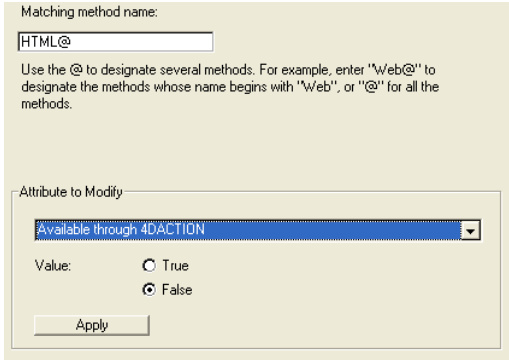
3 In the "Attribute to Modify" area, choose an attribute from the drop-down list, then click on the radio button corresponding to the value to be applied.

All of the different method attributes can be modified for a given batch:

- Invisible
- Available through 4DACTION
- Offered as a Web Service and Published in WSDL.

Note If the "Published in WSDL" attribute is set to True, it will only be applied to project methods already containing the "Offered as a Web Service" attribute.

In this example, the “Available through 4DACTION” option will be unchecked for all the methods that have names beginning with “HTML”:



4 **Click on Apply.**

The modification is applied instantly to all the project methods designated by the character string entered.

5 **Repeat the operation for each batch of methods and/or each attribute to be applied.**

6 **Click Done to close the dialog box.**

Using the Method Editor

4th Dimension’s Method editor works like a text editor. That is, you can enter and edit text in the editor.

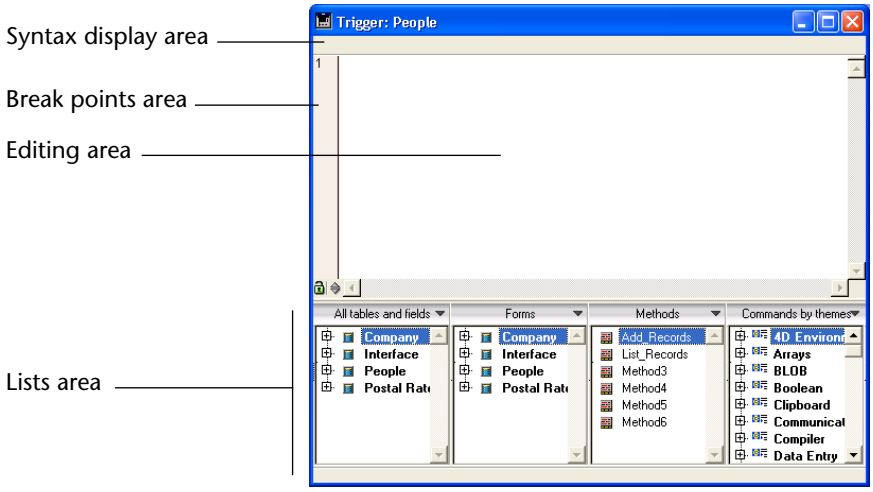
When you create a method with the Method editor, you write the method as a series of statements. You can also select components of the method from lists provided by the editor or via various shortcuts.

You can also drag and drop table names, field names, form names, project method names, commands, and constants from the Explorer, another method or another part of the same method.

It is possible to scroll through the method. You can enter up to 32,000 lines of command or 2 GB of text in a method.

Using and configuring the Method editor window

By default, a window of the Method editor consists of four areas: an editing area, a syntax display area, a lists area and a break points area. The figure below shows the default appearance of a Method editor window.



Editing Area

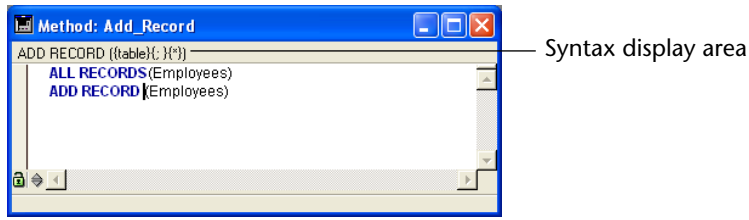
The editing area contains the text of the method. You enter and modify the method text in this area. The editor automatically indents method text for clear program structure. You can include comments inside the method text for reference.

You can customize the display of this area. To do this, refer to the following paragraphs.

For more information about the entry of code in this area, refer to the paragraph “Writing a Method,” page 559.

Syntax Display Area

This area displays the syntax of the 4th Dimension commands—the command name, followed by the parameter names— that you want to insert in your code. The syntax is displayed automatically when you enter a command manually. To “force” the display of the command syntax, insert a space after its name.



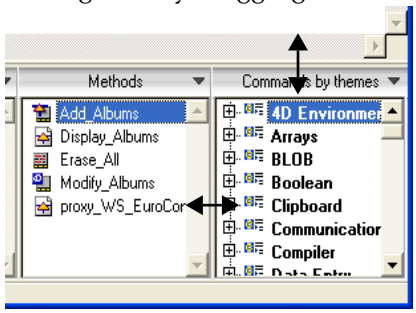
Note This mechanism requires the presence of the "4D Help" file in the active 4D folder of the machine (see the paragraph "Using Drag and Drop," page 81).

The syntax display area is also used to display potential syntax errors detected by 4th Dimension when the method is validated. For more information, refer to "Writing a Method" on page 559.

Lists area

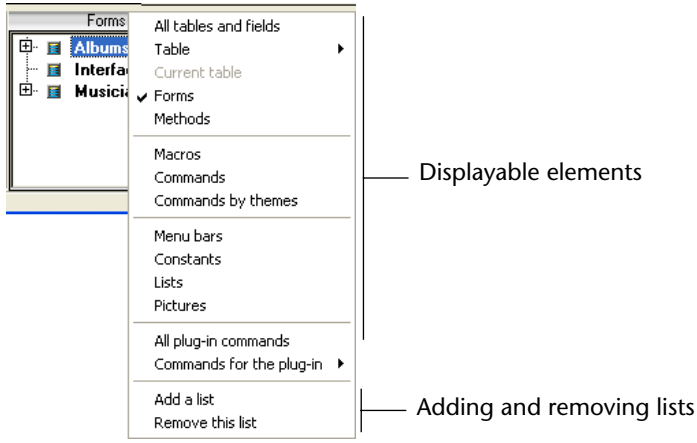
The lists area lets you display one or several lists of elements necessary for writing methods (commands, constants, forms, etc.). You can choose the number and contents of the lists displayed in the window.

By default, the Method editor displays four lists. You can enlarge or reduce the width of each list area by dragging one of its partitions. It is also possible to adjust the size of the list area in relation to that of the editing area by dragging the dividing line between them:



- **Double-clicking** on an element in a list causes it to be inserted into the editing area, at the location of the cursor.

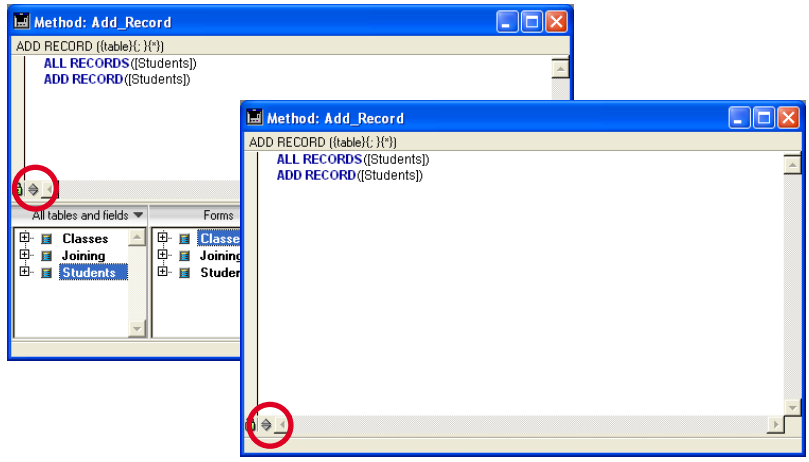
- To **modify the contents** of a list, click on the title area of the list concerned: a pop-up menu appears, enabling you to choose the type of element to be displayed.



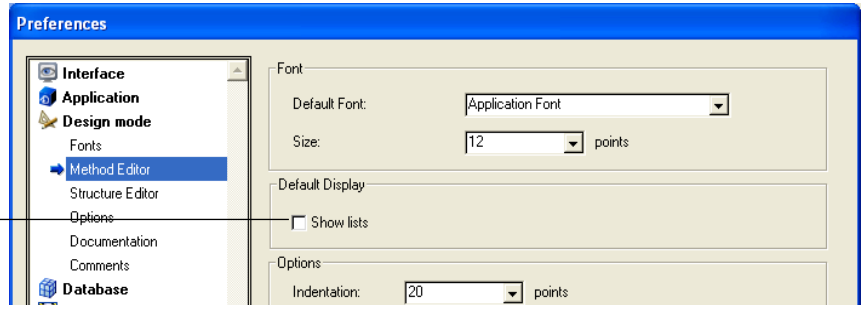
  For more information about the types of elements that can be displayed, refer to the paragraph “Description of the types of lists,” page 553.

- For **adding** or **removing a list**, click in the title area of one of the lists and choose the corresponding command in the pop-up menu (see below).
  Note that at least one list must be displayed in the editor window. The **Remove this list** command is disabled when you click on the last list. If you want to hide all the lists, you must use the database Preferences dialog box (see below).

- You can display only the editing area of the Method editor window. To do this, click on the icon located at the bottom left-hand corner of the editing area. The lists are then hidden. You can make them reappear by clicking again on the icon.



- You can also hide the lists by default. To do this, uncheck the **Show lists** option on the **Method Editor** page of the database Preferences dialog box (“Design mode” theme).



Option for displaying lists by default

Any open methods must first be closed then reopened for the modifications made in the Preferences dialog box to be taken into account.

Note You can save the parameters set in the Method editor window in the form of a template. For more information, refer to the paragraph “Save As Template,” page 555.

Break Points Area

This area allows you to directly insert break points next to specific instructions. Break points are useful when during the debugging phase of your programming. They stop the execution of your code at specific locations and display the debugger.

To insert a break point, click in the break points area at the location you want it to be. A red dot indicates the presence of a break point.

Note You can display the location of all the break points present in the database in the Runtime Explorer. For more information, refer to "Break and Catch Pages" on page 93.

To temporarily disable a break point or modify its properties, press the **Alt** key (Windows) or the **Option** key (MacOS) while clicking on the break point. When you click the break point, the break point property window is displayed.

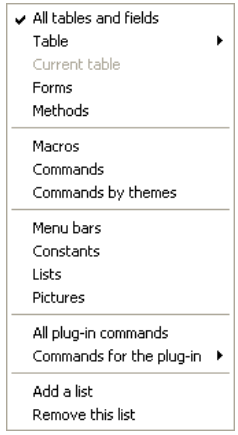
To delete a break point, click the red dot.

Note Break points remain at the location you defined it, even if you insert or delete a line.

For a complete description of break points, please refer to the *4th Dimension Language Reference* manual.

Description of the types of lists

You can display different lists of elements in the lists area of the Method editor window.



This paragraph details each type:

- **All tables and fields**: database table and field names in the form of a hierarchical list. When you insert a field name into the method by double-clicking on its name, 4th Dimension inserts it while respecting the syntax and adds the name of the table or subtable as the case may be.
- **Table** (submenu): field names of the table selected using the submenu.
- **Current table**: field names of the current table (available in triggers, form methods and object methods).
- **Forms**: database table and form names in the form of a hierarchical list. When you insert a form name into a method by double-clicking its name, 4th Dimension inserts it while respecting the syntax: the form name is inserted between quotes and is preceded by the name of the table and a semi-colon. For example: [Table];"Form"
- **Methods**: database project method names.
- **Macros**: macro names defined for the database, classified according to the order of the macro file.

Note The list of **keywords** present in previous versions of 4D is now handled using macros. For more information about macros, refer to the paragraph "Creating and using macros," page 577).

- **Commands**: 4th Dimension language commands in alphabetical order.
- **Commands by themes**: 4th Dimension language commands classified by theme in the form of a hierarchical list.
- **Menu bars**: menu bar names and numbers.
- **Constants**: 4th Dimension constants and any plug-ins, classified by theme in the form of a hierarchical list.
- **Lists**: names of lists.
- **Pictures**: names of pictures stored in the 4D Picture Library.
- **All plug-in commands**: commands for all the plug-ins installed in the database, classified by theme in the form of a hierarchical list.

- **Commands for the plug-in** (submenu): commands of a specific plug-in selected using the submenu.
  By default, the following plug-ins are available:
  - **4D Internet Commands**: This plug-in adds additional Internet functions to 4D, in particular for the management of e-mail.
  - **4D Chart**: 4D Chart is a chart editor built into 4th Dimension.
  - **OLE_Tools** (Windows only): OLE_Tools, supplied with the Windows version of 4D, is a plug-in for the management of OLE areas.

Note These plug-ins have their own separate documentation.

Note Except for the **Macros** element, all the lists are in alphabetical order.

Save As Template

You can save the parameters set in the Method editor window in the form of a "template." Once the template is saved, the parameters defined in it will be used for each new Method editor window that is opened.
The following parameters are stored in the template:

- Relative size of the editing and list areas
- Number of lists
- Location and contents of each list
- Relative width of each list.

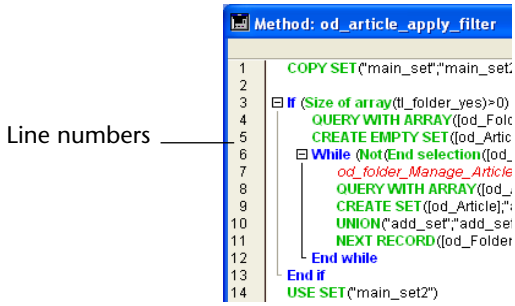
To save a Method editor window as a template, choose the **Save As Template** command in the **Method** menu or in the contextual menu of the Method editor:

Method editor contextual menu

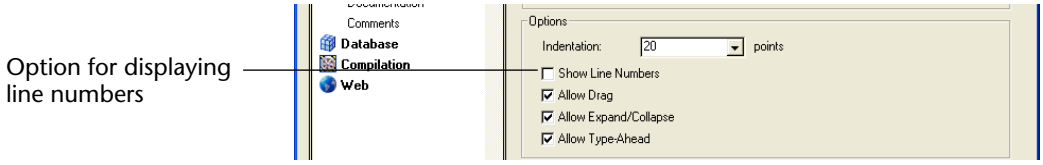
The template is saved immediately (no dialog box appears). It is stored in the preferences of the 4D application. If a previous template already exists, it is replaced.

Display of line numbers

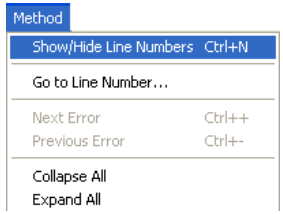
It is now possible to display the line numbers in each window of the Method editor:



The display of line numbers can be activated or deactivated by default using the **Show Line Numbers** option on the **Method Editor** page ("Design mode" theme) of the database Preferences dialog box:



It is also possible to modify this display separately for each window of the Method editor, using the **Show/Hide Line Numbers** command in the **Method** menu:



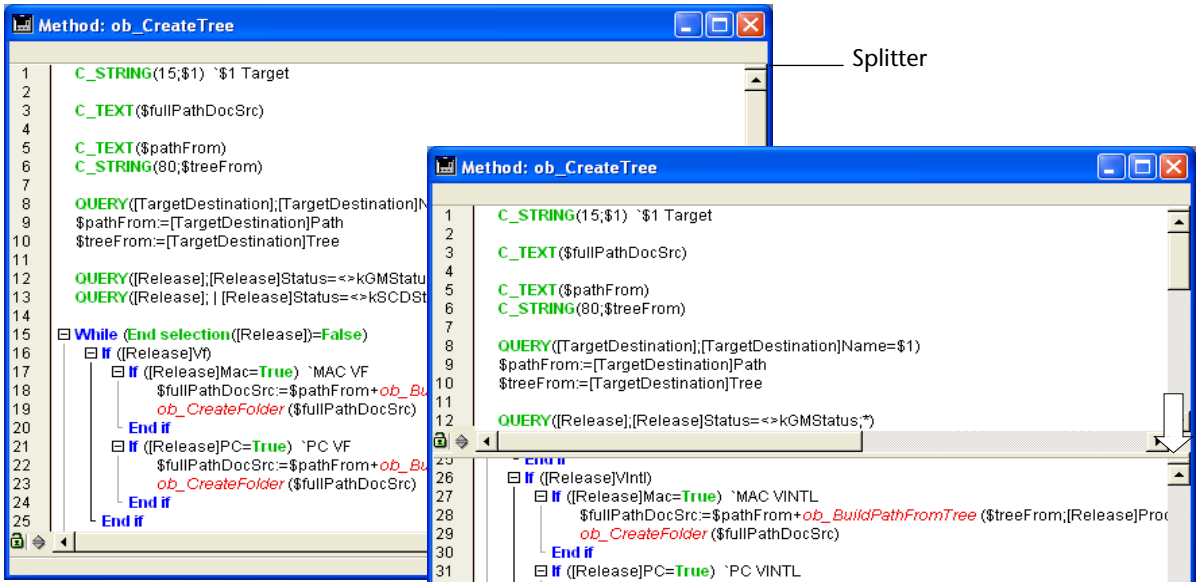
Displaying the line numbers makes it easier to find your way around in the window. The **Go to Line Number...** command in the **Method** menu also lets you take advantage of this display (see the paragraph "Go to Line Number," page 585).

Splitting the editor window

You can split the editing area into several horizontal panes. Once the editing area is split, you can view and scroll each part of the same method independently from the other parts. This is useful, for instance, when you want to keep the header of a method on screen — generally containing the description of the method, as well as any comments or declarations of variables. It also enables simultaneous viewing of two or more distant areas of the same method.

■ Creating panes

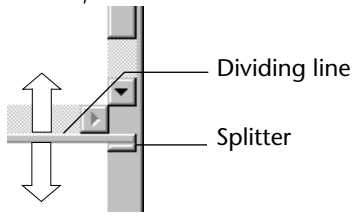
To create a viewing pane, click on the splitter located at the top of the window and slide it towards the bottom:



To create several panes, simply repeat the operation as many times as necessary. You can create a pane below an existing pane by clicking (once) on the splitter and dragging it downwards.

■ Resizing a pane

To resize a pane, slide one of its dividing lines upwards or downwards. *Be careful* not to confuse the dividing line of a pane with its splitter:



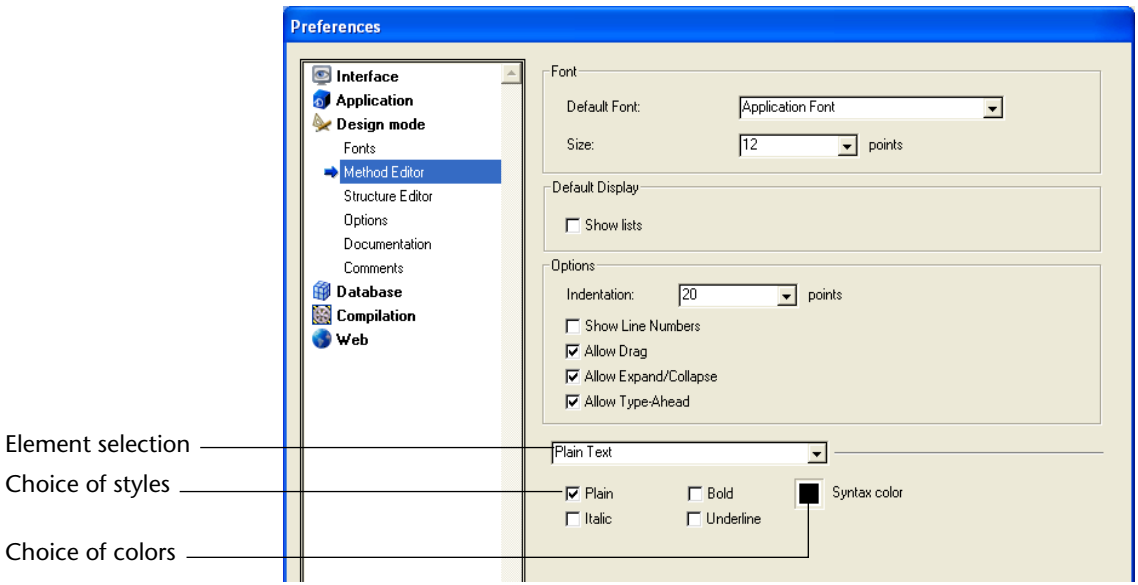
The splitter will not appear if the height of the pane is insufficient.

- **Removing a pane**
  To remove a viewing pane, slide its lower dividing line to the top of the window or double-click on this line.

Customizing styles and colors of syntax elements

In 4D, it is possible to assign a specific color to each type of element of the 4D language (fields, tables, variables, parameters, etc.). The combination of different colors and styles for the method elements can prove to be extremely useful for code maintenance.

Styles and colors are defined on the **Method Editor** page ("Design mode" theme) of the application Preferences dialog box:



Notes

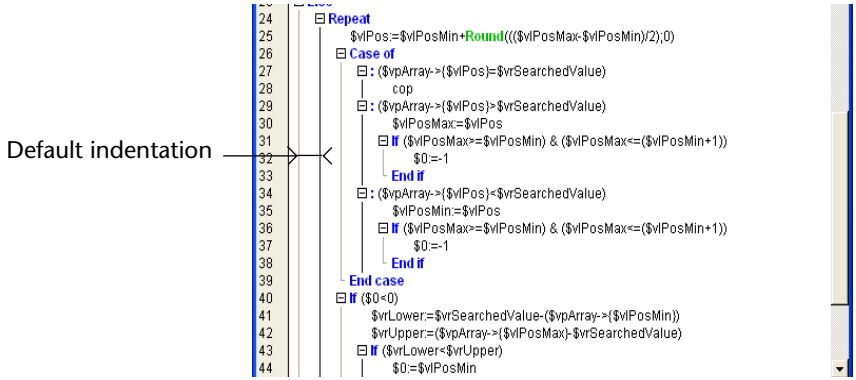
• The "Plain text" element type indicates all the texts not belonging to any other defined type (i.e., symbols, punctuation, literal constants, etc.).
• The "Parameters" element type indicates the method parameters ($0; $1, etc.).

The following styles are available:

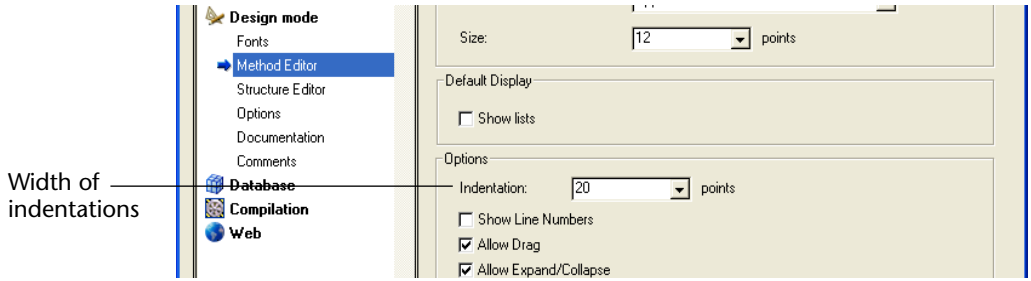
- **Normal**
- **Bold** (associated by default with 4D commands, keywords and plug-in commands).
- **Italic** (associated by default with plug-in commands and methods)
- **Underline** (associated by default with predefined constants).

Width of indentations

The code is automatically indented in order to reveal the code structure.



An option, located on the **Method Editor** page ("Design mode" theme) of the application Preferences dialog box, enables you to modify the indentation width:



The width must be defined in points (20 by default).
Modifying this default value can be useful if your methods contain complex algorithms with many levels of embedding. Narrower indentation can be used in order to limit horizontal scrolling.

Writing a Method

Writing a method is usually a combination of typing text, selecting components, and dragging elements from the Explorer or other windows. You can also use various type-ahead functions to speed the creation of methods.

The 4th Dimension Method editor provides basic syntax error-checking. Additional error-checking is performed when the method is executed.

You can customize the appearance and operation of the method editing area, for example by adding panes or displaying the line numbers.

Typing Text

4th Dimension uses standard text editing techniques for typing and editing in the Method editor. As you type, the characters appear at the location of the insertion point. You end each line by pressing the **Return** key (MacOS) or **Enter**[1] key (Windows).

Note To enter a numeric value in hexadecimal, type 0x (zero + "x"), followed by the hexadecimal digits.

The Method editor uses default display conventions (style, color) for command names, plug-in commands, methods, etc. You can modify these display conventions in the Preferences of the application (refer to the paragraph "Style and color of syntax elements," page 124). The font and font size can also be modified in the Preferences dialog box.

When you press the **Return**/**Enter** key, 4th Dimension evaluates the text of the line and applies the appropriate display format. 4th Dimension also indents each line to its proper level in relation to the preceding line when programming structures (for example If, End if) are used.

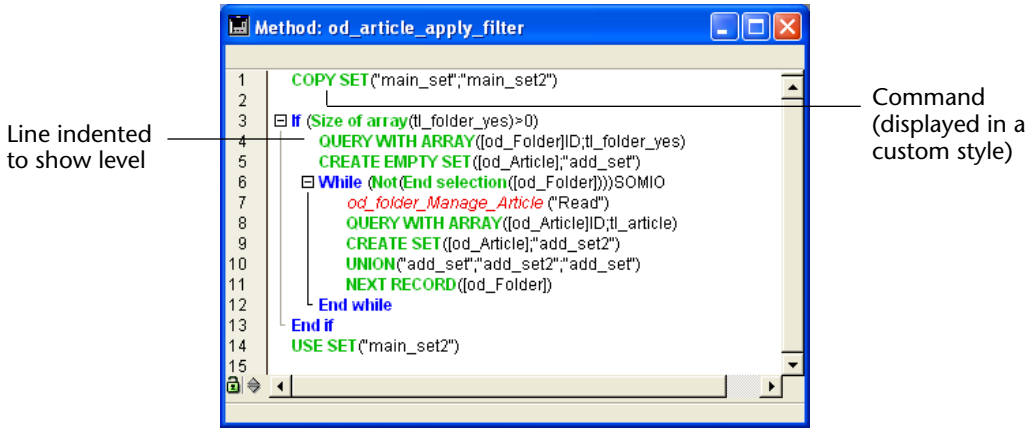
You can move the insertion point by clicking at the location you want. You can select words, whole lines, or several lines by dragging the I-beam over them.

You can use the arrow keys to quickly move from line to line. Using the arrow keys to move across several lines is quicker than clicking because the editor delays evaluating the line for errors.

Note The Method editor includes numerous navigation shortcuts. These shortcuts are listed in "Navigational Keyboard Shortcuts" on page 576.

1. The Enter key on the numeric keypad behaves differently than the Enter key on the main keyboard. Use the Enter key on the numeric keypad to force 4th Dimension to check the syntax of the line of code without moving the insertion point to the next line.

The following figure shows a method in the Method editor.



Brace Matching

The Method editor has a brace matching option in the **Method** menu that helps you balance braces, parentheses, quotes, and brackets. A check mark appears next to the selected option.

There are two levels of brace matching:

- **Small Brace Matching** affects only the opening and closing characters.
- **Big Brace Matching** affects the entire expression enclosed by the opening and closing characters.

When brace matching is active, 4th Dimension tries to find the matching brace, quote, or parenthesis when you type the closing character. When 4th Dimension finds matching characters, either the characters or the entire expression flashes.

For example, when you type

For ($i;1;**Records in selection** ([Line Items

and you press the closing bracket "]" to finish the table name expression, 4th Dimension will try to find the opening bracket. When it finds it, it flashes either the opening and closing brackets (Small brace matching) or the table name itself (Big brace matching).

As you continue to type the closing parentheses

For ($i;1;**Records in selection** ([Line Items]))

brace matching continues to provide feedback as you complete the arguments for the Records in selection function and the For keyword.

Turn off brace matching by choosing **No Brace Matching** from the **Method** menu.

Adding Method Objects by drag and drop

4th Dimension allows you to use the drag and drop mechanism when writing methods. It is possible to drag and drop elements:

- from the Explorer,
- within the same method,
- between two methods.

- **Drag and drop from the Explorer**

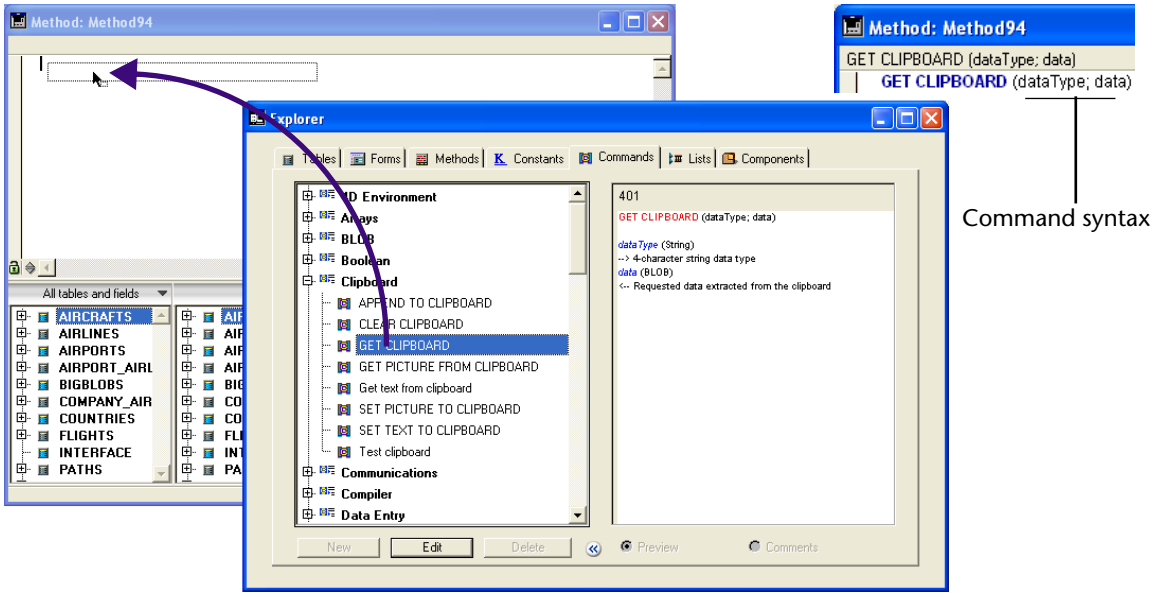
  From the Explorer, you can drag and drop:

- Table names and field names from the Tables page,
- Table names and form names from the Forms page,
- Project methods and form names from the Methods page,
- Constants from the Constants page,
- 4th Dimension commands from the Commands page.

  When you drag and drop a component, 4th Dimension always uses the correct syntax for the component. For example, if you drag the field name “First Name” from the [People] table, it appears in the Method editor as “[People]First Name”. Similarly, if you drag the Form name “Input” from the People table, it appears in the Method editor as “[People];"Input"”.

  When you insert a command by dragging it from the Explorer, it appears with its syntax (which consists of all of its parameters) in the Method editor.

Of course, you use the syntax that you need to adapt to your usage. This feature reminds you of the parameters that the command expects:



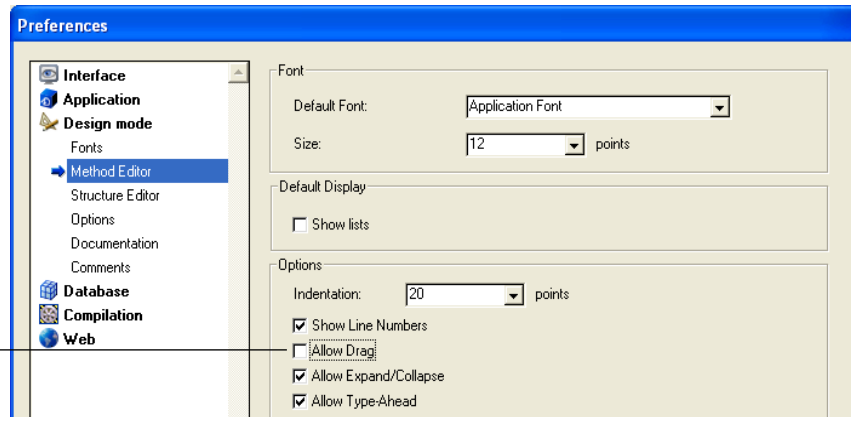
You can choose not to insert the syntax when dragging a command by pressing the **Alt** key (Windows) or the **Option** key (MacOS) when dragging the command.

- **Drag and drop within a method or between two different methods**

the Method editor, the drag-and-drop mechanism is activated as soon as a portion of text is selected. The cursor changes as shown here to indicate that the selection can be dragged and dropped :

- *Copying the selection*
  By default, the drag-and-drop mechanism moves the selected text. In order to copy it, hold down the **Ctrl** key (Windows) or the **Option** key (MacOS) during the operation.

- *Disabling drag and drop*
  The drag-and-drop mechanism inside a method or between two methods can be enabled/disabled using the **Allow Drag** option (checked by default) located in the application Preferences dialog box, on the **Method Editor** page ("Design mode" theme):

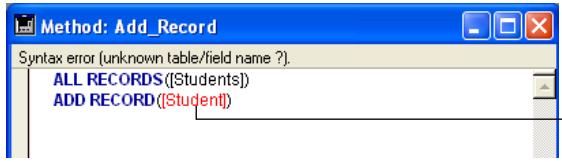


Disabling of drag and drop

Note Inserting objects from the 4D Explorer using drag and drop remains possible even when the drag-and-drop mechanism is disabled inside the Method editor.

Checking and Correcting Syntax Errors

4th Dimension automatically checks the method syntax to see if it is correct. If you enter text or select a component that is not syntactically correct, 4th Dimension specifies the error in the syntax display area and indicates the incorrect expression, as shown in the following illustration.



Location of the error (here the "s" of the [Students] table is missing)

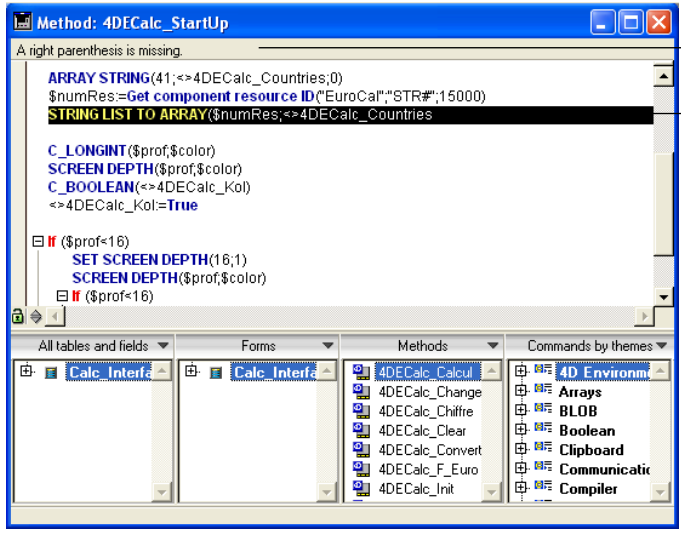
The style used to display the incorrect expressions can be modified in the Preferences of the application (see the paragraph "Style and color of syntax elements," page 124).

You can immediately check the syntax of the current line (without advancing to the next line) by pressing the **Enter** key (MacOS) or **Enter** key on the Windows numeric keypad. 4th Dimension evaluates the line, formats it, marks any errors, and places the insertion point at the end of the line.

When a line of a method is marked as having improper syntax, check and fix the entry. If the line is now correct, 4th Dimension returns to the standard style.

The validation of an entire method is done when you close the window. You can also force validation by pressing the **Enter** key.

When the method is validated, 4th Dimension checks for basic syntax errors and for the structure of the statements (**If**, **End if** and so on). When an error is detected, a message is displayed in the syntax display area and 4th Dimension highlights the line that contains the mistake.

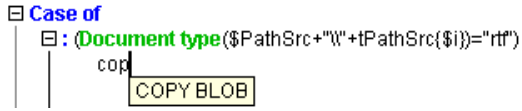


The Method editor can check only for obvious syntax errors (misspellings and the like). It does not check for errors that occur only during execution. Execution errors are caught by 4th Dimension when the method is executed. 4th Dimension provides a debugger for handling and correcting these errors. For information about the debugger, refer to the *4th Dimension Language Reference* manual.
The compiler also provides indispensable help for detecting errors. For more information about the compiler, refer to the chapter “Compiling a Database,” page 689.

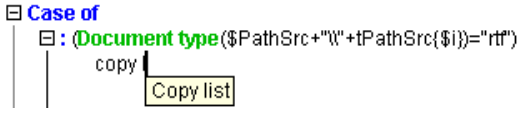
Type-ahead function

The Method editor has a “type-ahead” function. 4th Dimension automatically displays propositions based on the first few characters typed.

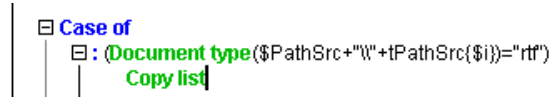
In the example given below, typing the string “cop” causes the display of a tip containing the first 4D command (by alphabetical order) beginning with this string:



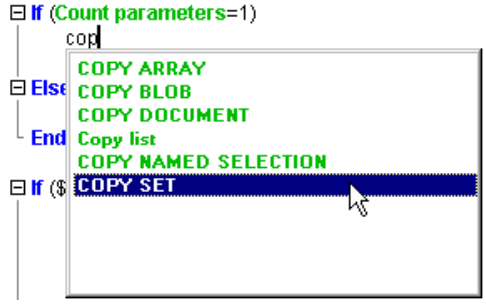
The value proposed in the tip is updated when additional characters are typed:



When the characters typed correspond to only one possible value, this value is inserted by pressing the **Tab** key:

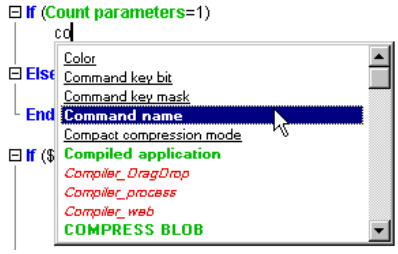


Otherwise, pressing the **Tab** key displays a list of all the words that begin with the characters typed:



The list is in alphabetical order. You can choose a value in the list by double-clicking on it or using the ↑ and ↓ arrow keys to scroll among the selections and then hitting the **Carriage return**. You can also press the **Esc** key and continue typing.

If the characters typed correspond to different types of objects, the list displays them in their current style:



The following types of objects can be displayed:

- 4D command
- User method
- Table name
- Field name
- Constant
- Plug-in command
- 4D keyword
- Macros.

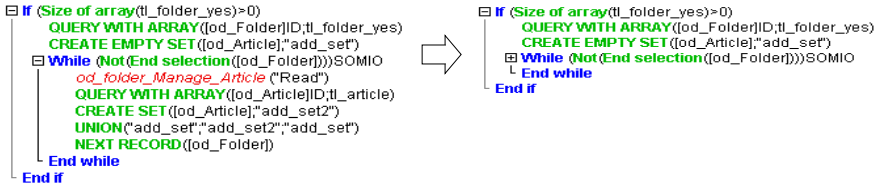
Note The macro names are displayed between < >. For more information about macros, refer to the paragraph "Creating and using macros," page 577.

- **Disable Type-Ahead**
  The type-ahead functions can be enabled/disabled using the **Allow Type-Ahead** option located in the application Preferences dialog box, on the **Method Editor** page ("Design mode" theme).

- **Using the "Wildcard" (@)**
  Former versions of 4th Dimension used the wildcard character "@" as a shortcut when inserting object names. This mechanism can still be used, even though it is less powerful than the type-ahead mechanism. When you type several characters followed by the "@" character, 4th Dimension looks for element names beginning with these characters and completes the entry with the first name it finds that corresponds to the criteria. It is essential to enter enough characters to differentiate the desired word from similar ones since in the event of there being more than one possible solution, 4th Dimension may complete the entry by one of the other possibilities.

Expand/Collapse

4D code located inside loops and conditions can now be collapsed or expanded, in order to facilitate the reading of methods.



The operation of this new function is similar to that of hierarchical lists: clicking on the ⊟ icon causes the code included in the condition or loop to be collapsed. The ⊞ icon indicates a portion of collapsed code; clicking on this icon causes the portion of collapsed code to be expanded.

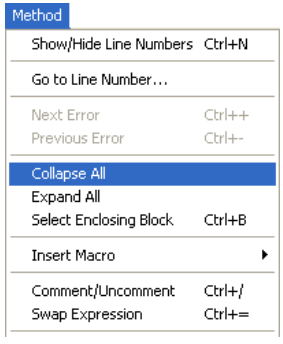
Note Conditions and loops (**If**, **Case of**, etc.) are defined with the help of macros. For more information, refer to the paragraph "Creating and using macros," page 577.

When a portion of code is collapsed, it can no longer be modified. It is automatically expanded when you enter or delete a character in the condition or loop concerned.

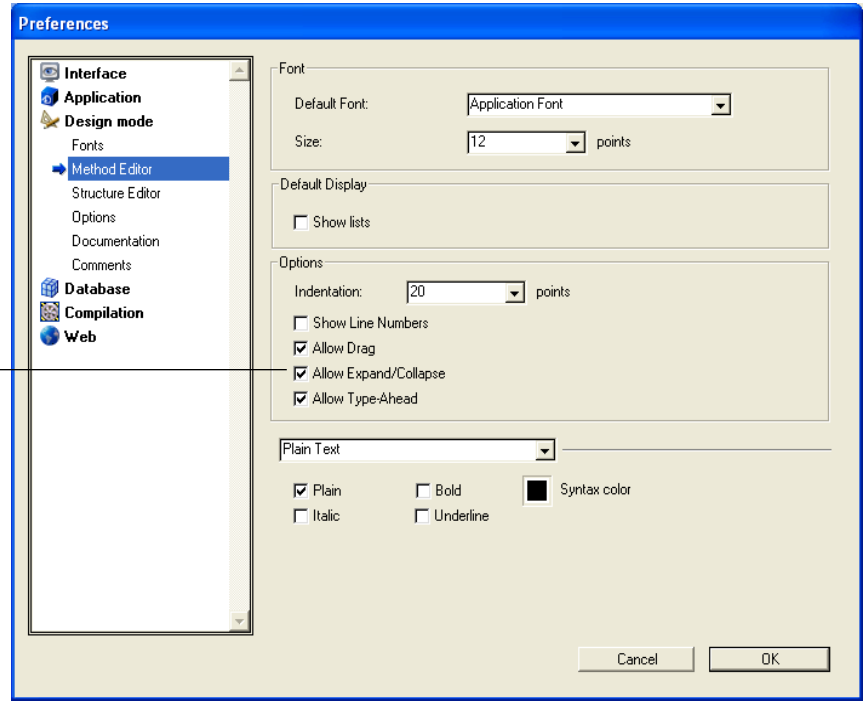
A collapsed portion of code can be selected, copied, pasted or deleted. All the lines included therein will be copied, pasted or deleted respectively.

Note When a portion of code is pasted, it is automatically expanded.

It is possible to expand/collapse all of the loops and conditions of a method using the **Collapse all** and **Expand all** commands in the **Method** menu or in the Method editor contextual.



The display of loops and conditions in the form of hierarchical lists can be activated or deactivated using the **Allow Expand/Collapse** option on the **Method Editor** page (“Design mode” theme) of the database Preferences dialog box:



Activation of code hierarchical display option

This option is checked by default.

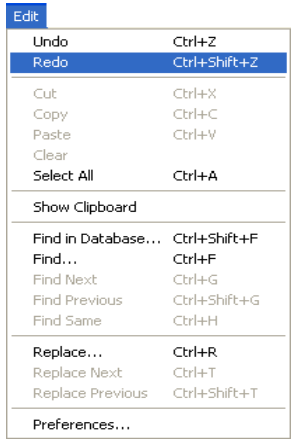
Note The display mode is not taken into account during printing of methods: as in previous versions of 4D, the source code is always expanded and vertical lines outline the conditions and loops.

Multiple Undo/Redo

The Method editor of 4D 2003 enables multiple undo/redo operations. All the actions performed (text entry, deletion, copy/paste, etc.) are stored sequentially in memory and can be undone in the reverse order of their execution. Likewise, each action undone can be redone.

4th Dimension thus records the last 20 actions carried out.

The **Undo/Redo** commands are available in the **Edit** menu and the Method editor contextual menu, as well as in the toolbar.

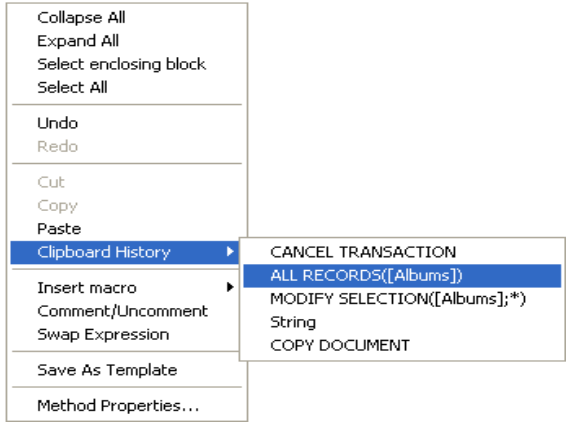


Multiple copy-paste and numbering of clipboards

In addition to the standard copy-paste operation, 4th Dimension 2003 proposes two additional functions that let you work with the contents of different clipboards:

- The program stores in memory the last 10 “copy” or “cut” actions that were performed in the Method editor during the current session. Each of the different contents saved in this way can be reused at any time.

  To do this, use the **Clipboard History** command of the Method editor contextual menu:



  The first few words of the copied or cut elements are displayed. Selecting an element causes it to be inserted at the current location of the cursor.

- The 10 current clipboards are numbered and can be employed directly using a keyboard shortcut:
  - to *copy* the selection into a specific clipboard, use **Ctrl+Shift+1** to **9** (Windows) or **Control+Shift+1** to **9** (MacOS).
  - to *paste* the contents of a specific clipboard, use **Ctrl+1** to **9** (Windows) or **Control+1** to **9** (MacOS).

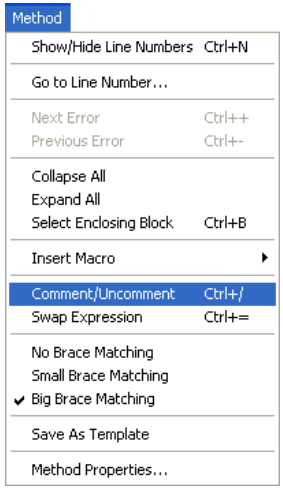
Comment/Uncomment

Comments are inactive lines of code. These lines are not interpreted by the program (4th Dimension does not apply any particular style within comments) and are not executed when the method is called.
To create a comment, you just need to insert the ` character at the beginning of a line.

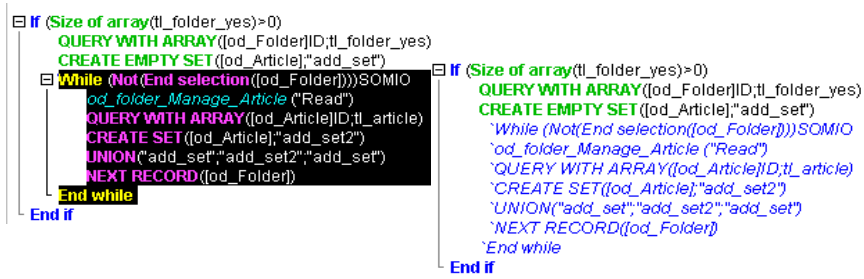
The length of comments is limited to the maximum size of a line, which is 32,000 characters.

The **Comment/Uncomment** command is used to mark a group of selected lines of code as comments, or, on the contrary, to remove the comment characters from a selection.

The **Comment/Uncomment** command is available in the **Method** menu as well as in the Method editor contextual menu.



To use this command, select the code to be marked as commented, then select the **Comment/Uncomment** command:



When the selection contains only active code, the **Comment** command is applied.

When the selection includes both active code and commented lines, an additional comment character (`) is added to the latter; in this way, they will retain their initial commented status if the line is subsequently "uncommented."

When the selection contains only commented lines, the **Uncomment** command is applied.

Note The **Comment/Uncomment** command only operates with full lines — it cannot be used to comment only part of a line.

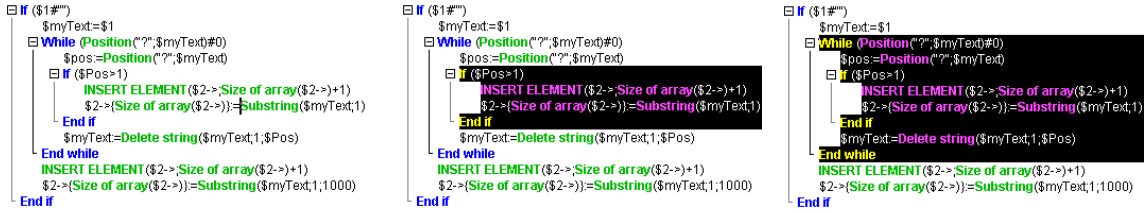
Selection of enclosing block

The **Select Enclosing Block** function is used to select the "enclosing block" of the code containing the insertion point. The enclosing block can be defined by:

- quotes,
- parentheses,
- a logical structure (If/Else/End if, While/End while, Repeat/Until Case of/End case), or,
- braces.

If a block of text is already selected, the function selects the enclosing block of the next highest level and so on, until the entire method is selected.

In the following example, the **Select Enclosing Block** function is applied two times consecutively in order to select the desired block of code.



Pressing **Ctrl+Shift+B** (Windows) or **Command+Shift+B** (MacOS) enables you to reverse this operation and deselect the last enclosing block selected.

Note If the insertion point is placed in an **If** or **Else** type structure, the enclosing block will be the one containing, respectively, the **If** or **Else** statement.

Swap expression

The **Swap expression** function can be used to reverse the arguments of an expression assigning a value. For instance,

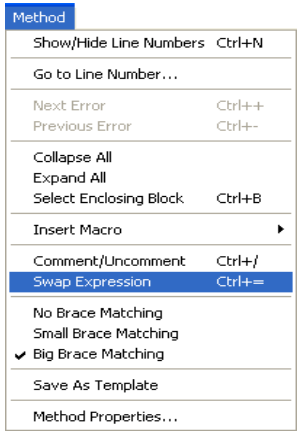
variable1:=variable2

becomes

variable2:=variable1

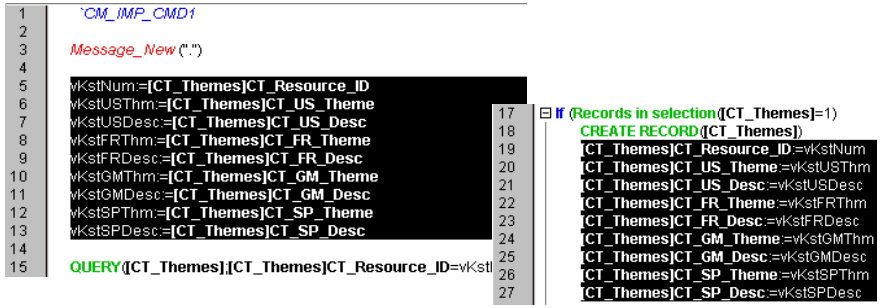
This function is extremely useful for reversing a set of assignments used to get or set properties, or to correct input errors.

To use this function, select the line(s) to be modified, then choose the **Swap expression** command in the **Method** menu, or in the contextual menu of the area:



Within the selection, only the lines assigning a value will be modified.

In the import method shown below, the variable assignment area located at the beginning of the method is copied, then the expressions it contains are reversed using the **Swap expression** command:



Managing of long strings

In the Method editor of 4th Dimension, character strings are limited to 80 characters; however, the entry of long strings is managed automatically: the editor divides strings that are too long and inserts the necessary syntax elements at the time of line validation:

Input:

```
ALERT("This character string is really much too long to be kept in its original format; it will be divided automatically by the method editor.")
```

After validation:

```
ALERT("This character string is really much too long to be kept in its original format; "+"it will be divided automatically by the method editor.")
```

Use of escape sequences

The Method editor allows you to use *escape sequences* (also called *escape characters*). An escape sequence is a sequence of characters that can be used to replace a "special" character.

The sequence consists of a backslash \, followed by a character. For instance, \t is an escape sequence for the Tab character. Escape sequences facilitate the entry of special characters: the previous example (\t) replaces the entry Character(Tab).

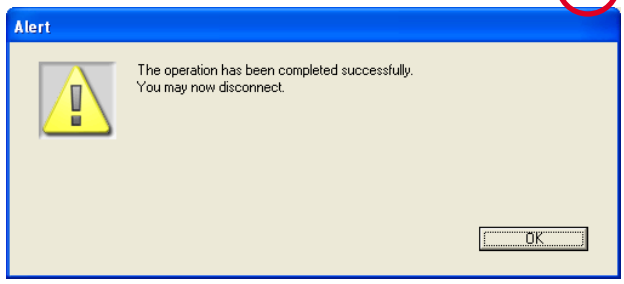
In 4th Dimension, the following escape sequences can be used:

Escape sequence	Character replaced
\n	LF (Line feed)
\t	HT (Tab)
\r	CR (Carriage return)
\\	\ (Backslash)
\"	" (Quotes)

Note It is possible to use either upper or lower case in escape sequences.

In the following example, the *Carriage return* character (escape sequence \r) is inserted in a statement in order to obtain the dialog box shown below:

```
ALERT("The operation has been completed successfully.\rYou may now disconnect.")
```



The same statement could have been written as follows:

```
ALERT("The operation has been completed successfully."+Char(Carriage return)+"You may now disconnect.")
```

Warning: The \ (backslash) character is used as a separator in pathnames under Windows. In general, 4th Dimension will correctly interpret Windows pathnames entered in the Method editor by replacing the single backslash \ with a double backslash \\. For instance, C:\Folder will become C:\\Folder.

However, if you write "C:\MyDocuments\New", 4th Dimension will display "C:\\MyDocuments\New". In this case, the second backslash \ is interpreted incorrectly as \N (an existing escape sequence). You must therefore enter a double backslash \\ when you want to have a backslash in front of a character used in one of the escape sequences recognized by 4th Dimension.

Navigational Keyboard Shortcuts

Helpful keyboard shortcuts to navigate the code are available in 4D's Method editor[1].

Windows	MacOS	Action
[Shift]+[→]		Create and enlarge the selection, character by character, to the right, or Reduce the selection, character by character, from the left
[Shift]+[←]		Reduce the selection, character by character, from the right or Create and enlarge the selection, character by character, to the left
[Shift]+[↓]		Create and enlarge a selection, line by line, from the top to the bottom
[Shift]+[↑]		Create and enlarge a selection, line by line, from the bottom to the top
[Ctrl]+[Shift]+[→]	[Command]+[Shift]+[→]	Create and enlarge the selection, word by word, from the right
[Ctrl]+[Shift]+[←]	[Command]+[Shift] +[←]	Reduce the selection, word for word, from the right, or create and enlarge the selection, word by word, from the left
[Ctrl]+[→]	[Command]+[→]	Move the insertion point, word by word, from left to right
[Ctrl]+[←]	[Command]+[←]	Move the insertion point, word by word, from right to left
[Home]		Place the insertion point at the beginning of the line
[End]		Place the insertion point at the end of the line
[Ctrl]+[Home]	[Command]+[Home]	Place the insertion point at the beginning of the method
[Ctrl]+[End]	[Command]+[End]	Place the insertion point at the end of the method
[Shift]+[Home]		Select all the characters in the line that are to the left of the cursor
[Shift]+[End]		Select all the characters in the line that are to the right of the cursor

1. These shortcuts are also available in all of 4th Dimension's dialog boxes that contain data entry areas.

Windows	MacOS	Action
[PgUp]		Scroll the contents of the method, page by page, from the bottom to the top (doesn't modify the insertion point)
[PgDn]		Scroll the contents of the method, page by page, from the top to the bottom (doesn't modify the insertion point)

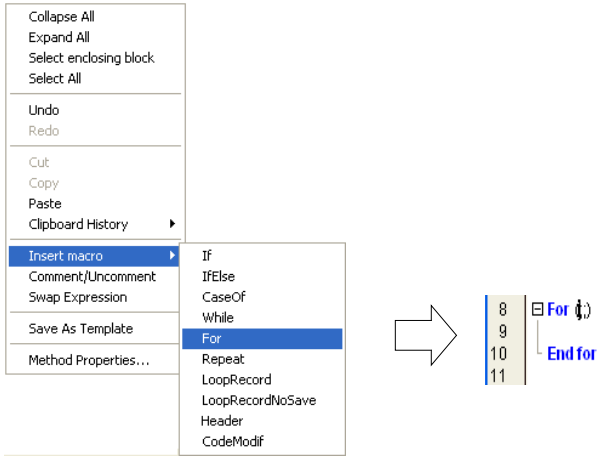
Creating and using macros

You can use macro-commands in your methods. Using macro-commands saves a lot of time during method entry.

What is a macro?

A macro-command is a section of 4D code that is permanently accessible and that can be inserted anywhere in your methods, whatever the type of database open. Macros can contain all types of 4D text, commands and constants, as well as special tags which are replaced at the time of macro insertion by values derived from the method context. For instance, a macro may contain the tag <method_name/>; at the time of macro insertion, this tag will be replaced by the name of the current project method.

Example of macro insertion



Macros are stored in an XML format (text) file. They can be placed in a Method editor list; they can also be called using the contextual menu of the editor or using the type-ahead function.

4th Dimension macros are written in XML format. You can use the 4D default macro file "as is" or modify it.

Location of macros

Macros are specific to each machine and are available for all the 4th Dimension applications running on the machine. All the macros are stored in a single text file named "Macros.xml" located in the **active 4D folder** of the machine.

Note The location of the active 4D folder varies according to the operating system used. For more information, refer to the *4D Product Line Installation Guide*.

Default macros

4th Dimension offers a set of default macros corresponding, in particular, to the list of keywords in previous versions of 4D. These macros are included in the default "Macros.xml" file, created in the active 4D folder of the machine during the initial startup of 4th Dimension.

Note 4th Dimension allows you to embed programming structures (If/While/For/Case of) up to a "depth" of 512 levels.

You can modify this file subsequently as desired (see the following paragraph). In the event of problems with this file, it can be deleted and 4th Dimension will re-create it on the next startup.

Adding customized macros

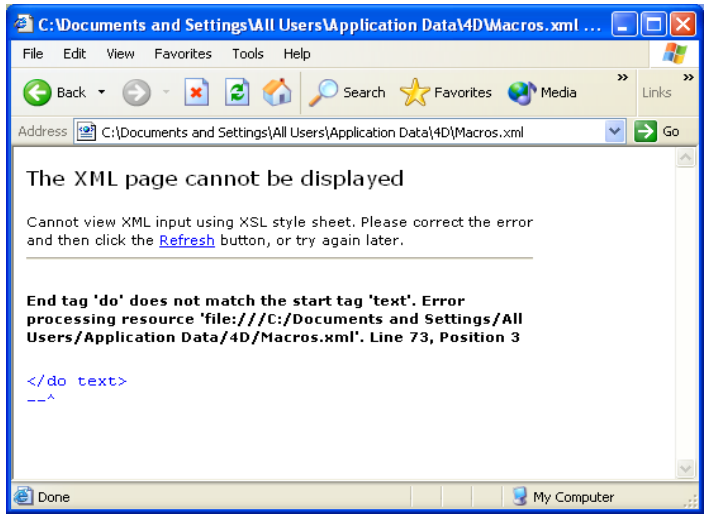
You can add customized macros in the "Macros.xml" file using a standard text editor or by programming.

The macros file can be open while using 4th Dimension. The list of available macros is updated on each event activating 4th Dimension. For instance, it is possible to bring the text editor to the foreground, modify the macro file, then return to the method: the new macro is then available in the Method editor.

Empty or erroneous macros are not displayed.

- **Checking the syntax of customized macros**
  Since the language used to define 4th Dimension macros is XML, you can make sure that the "Macros.xml" file does not contain any syntax errors simply by opening it with a Web browser. Web browsers can display the contents of XML files in the form of a hierarchical list and parse their syntax.

If an error is detected (for instance, a missing end tag), the browser will indicate its location:



Note You can also "validate" the Macros.xml file, i.e., check for conformity with its Document Type Declaration (DTD), supplied by 4D. For more information, refer to Appendix C on page 761.

Syntax of 4D macros

4D macros are built using customized XML tags called "elements."

Some tags indicate the start and end of the definition (double tags of the type <tag> </tag>), others are replaced by insertion context values (<tag/>).

In conformity with XML specifications, some element tags can include **attributes**. Unless otherwise indicated, these attributes are optional and a default value is used when they are omitted. The syntax of elements with attributes is as follows:

- double tags: <tag attribute="value"> </macro>
- single tags: <tag attribute="value"/>

If the element accepts several attributes, you can group them in the same line of command, separated by a space:
<tag attribute1="value" attribute2="value" attribute3="value"... >

Here is the list of tags and their mode of use:

Element tags	Description
<**macros**> </**macros**>	Start and end of macro file (mandatory tag).
<**macro**> </**macro**>	Start and end of the definition of a macro and its attributes. ***Attributes***: • **name**: Name** of macro as it appears in menus and Method editor lists (mandatory attribute). • **type_ahead_text**: Character string** to be entered to call the macro using the type-ahead function*. • **in_menu**: Boolean indicating whether the macro can be called using the contextual menu*. Values = "true" (default) or "false". • **type_ahead**: Boolean indicating whether the macro can be called using the type-ahead function*. Values = "true" (default) or "false".
<**selection**/>	Tag replaced by the selected text when the macro is inserted. The selection may be empty.
<**text**> </**text**>	Start and end of code that must be inserted in the method. A carriage return will be added before and after the code.
<**caret**/>	Location of the insertion point in the code after the macro has been inserted.
<**user_4D**/>	Tag replaced by the name of the current 4D user.
<**user_os**/>	Tag replaced by the current system user name.
<**method_name**/>	Tag replaced by the current project method name.
<**date**/>	Tag replaced by the current date. ***Attribute***: • **format**: 4D format used to display the date. If no format is defined, the default format is used. Values = number of 4D format (0 to 8).
<**time**/>	Tag replaced by the current time. ***Attribute***: • **format**: 4D format used to display the time. If no format is defined, the default format is used. Values = number of 4D format (0 to 6).
<**clipboard**/>	Tag replaced by the contents of the clipboard. ***Attribute***: • **index**: Clipboard to be pasted. Values = number of the clipboard (0 to 9).

* Macros can be called using the contextual menu of the Method editor or using the type-ahead function (see the following paragraph).

** Macros can be called using the contextual menu of the Method editor or using the type-ahead function (see the following paragraph.).

Here is an example of a macro definition:

Content of macro	Comments
<macros>	*Start of macros XML file*
<macro name="RecordLoop">	*Start of macro definition and name*
<text>	*Start of macro code*

For($i;1;Records in selection(<Selection/>))
SAVE RECORD(<Selection/>)
NEXT RECORD(<Selection/>)
End for
	The <Selection/> tag will be replaced by the selected code in the 4D method at the time of macro insertion (for instance, a table name)

</text>	*End of macro code*
</macro>	*End of macro definition*
</macros>	*End of macros XML file*

Calling macros

By default, macros can be called using the contextual menu of the Method editor, the type-ahead function, or a specific list at the bottom of the Method editor window.
Note that for each macro it is possible to restrict the possibility of calling it using the contextual menu and/or the type-ahead function.

- **Contextual menu**
  By default, all macros can be called via the contextual menu of the Method editor, using the **Insert macro** hierarchical command. The in_menu attribute of the <macro> tag is used to define whether or not the macro will appear in this menu.

  In the contextual menu, macros are displayed in the order of the "Macros.xml" file. It is therefore possible to change the order by modifying this file.

- **Type-ahead**
  By default, all macros are accessible using the type-ahead function (see the paragraph "Type-ahead function," page 566). The macro replaces the entered text.
  The type_ahead attribute of the <macro> tag can be used to exclude a macro from this type of operation.

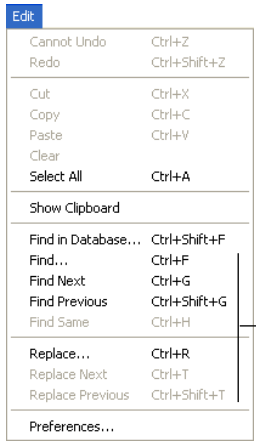
Note If the macro contains the <selection/> tag, it will not appear in the type-ahead pop-up window.

In order to distinguish them from other types of objects, macro names appear surrounded by the < and > characters in the type-ahead pop-up window.

- **Method editor list**
  You can display your macros in a list of the Method editor (see the paragraph “Lists area,” page 550). Simply double-click on the name of a macro in the list in order to call it.
  It is not possible to exclude a specific macro from this list.

Find and replace

The Method editor has specific find and replace functions that apply to the current window. The Find/Replace commands for methods are located in the **Edit** menu of 4th Dimension:

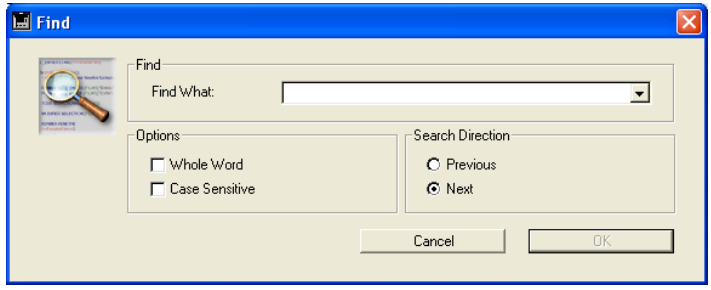


Find and replace functions for methods

Note The **Find in Database...** command enables an overall search in the database. It is not specific to the Method editor but may be used to search for a value among all the methods. For more information about this command, refer to the paragraph “Searching in the Database,” page 94.

Find

Selecting the **Find...** command displays the following dialog box:



The search defined in this dialog box will be performed in the method located in the foreground.

- The “Find What:” area enables you to enter the string of characters to be searched for. This area is a combo box that stores the last 15 character strings that have been searched for or replaced during the session. If you highlight text before choosing the **Find...** command, it will appear in this area. You can then either use this text or replace it with another.
- The **Whole Word** option is used to limit the search to exact occurrences of the word being searched for. When this option is checked, for instance, a search for “client” will not find either “clients” or “myclient.” By default, this option is not checked; therefore, a search for “var” will find “Myvar,” “variation,” etc.
  Be careful, unlike the **Whole Object Name** option of the Find in Database dialog box, the **Whole Word** option does not take *object names* into account. For example, with this option, searching for the string “My” in a method will find the “My Variable” variable. This is not the case for an overall search using the **Whole Object Name** option, where the same result will not be found in the context of the above example since the whole object name (of the variable found previously) is “My Variable” and therefore does not correspond exactly to the string entered (“My”).
- The **Case Sensitive** option is used to take the case of characters as they were entered in the “Find What:” area into account. For instance, a search for “MyVar” will not find “myVar.”
- The **Previous/Next** radio buttons are used to define the direction of the search: towards the beginning or end of the current method, starting from the initial location of the cursor.

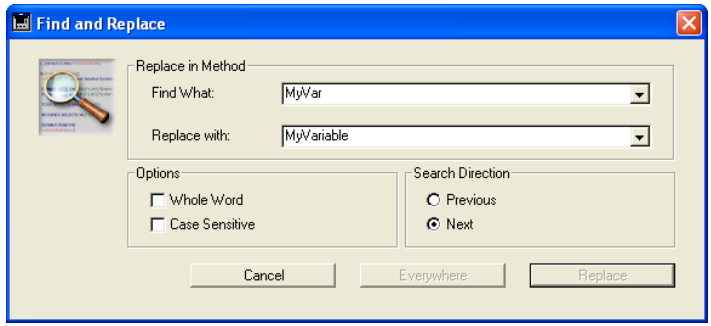
The **OK** button is used to launch a search. 4th Dimension begins searching from the current text insertion point and continues to the end of the method. The first element corresponding to the defined criteria is thus selected in the Method editor window. It is then possible to continue the search using the **Find Next** and **Find Previous** commands of the **Edit** menu.

Find Same

The **Find Same** command is used to find character strings identical to the one selected. This command is only active if you have selected at least one character in the Method editor.
The search carried out is of the “Find Next” type in the current method.

Find/Replace

The **Replace** command displays the following dialog box:



- The “Find What:” area is used to define the character string or the expression to be searched for. As in the Find dialog box, this area is a combo-box that stores the last 15 character strings searched for. If you highlight text before choosing the **Replace** command, it will appear in this area.
- The “Replace with:” area is used to define the character string that will replace the one defined above. This area is also a combo-box storing the last 15 character strings that have been searched for or replaced.
- The **Whole Word** option is used to find/replace only character strings that correspond exactly to the string entered. In this case, for instance, a search for “client” will not find the strings “clients” or “myclient,” etc.
- The **Case Sensitive** option is used to find/replace only character strings having the same case as that of the entered string. For instance, a search for “MyVar” will not find “myVar.”
- As in the Find dialog box, the **Previous** and **Next** buttons are used to define the direction of the search: towards the beginning or end of the current method, starting from the initial location of the cursor.

  The **Replace** button is used to launch the search and replace the first occurrence found. 4th Dimension begins searching from the current text insertion point and continues to the end of the method. It is then possible to continue finding/replacing using the **Replace Next** and **Replace Previous** commands of the **Edit** menu.

  The **Everywhere** button is used to directly replace all the occurrences corresponding to the search criteria in the open method.

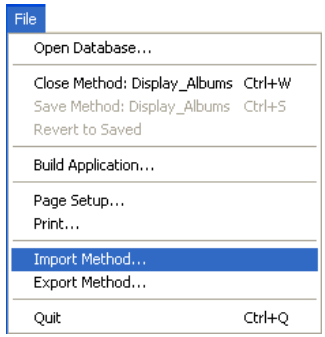
Go to Line Number

This command opens a dialog box where you can indicate the line number you want to find. When you click **OK**, the editor finds and highlights that line in the method. This type of search is useful when used in conjunction with the compiler, which flags runtime errors by the line number in which they occur.

Note You can choose whether or not to display lines numbers in the Method editor window (see the paragraph "Display of line numbers," page 556).

Importing and exporting methods

4th Dimension allows importing and exporting, in the form of a file, of database, project and object methods, as well as triggers. These commands are found in the **File** menu:



When you select the **Export method** command, a standard file saving dialog box appears, allowing you to choose the name, location and format of the export file (see below).
As with printing, exporting does not take the collapsed state of code structures into account and the entire code is exported.

When you select the **Import method** command, a standard file opening dialog box appears, allowing you to designate the file to be imported.
Importing replaces the selected text in the method. To replace an existing method by an imported method, select the entire contents of the method before carrying out the import.

Note The import/export function is multi-platform: a method exported under MacOS can be imported under Windows and vice versa; 4th Dimension handles the conversion of characters when necessary.

File formats

4th Dimension can export and import methods in two formats:

- **4D method** (extension ".c4d" under Windows): in this format, methods are exported in encoded form. The names of objects are tokenized. This format is used in particular for exchanging methods between 4th Dimension applications and plug-ins in different languages. Conversely, it is not possible to display them in a text editor.
- **Text** (extension ".txt" under Windows): in this format, methods are exported in text-only form.
  In this case, the methods are readable using a standard text editor. However, the languages of the 4D applications used for export and import must be identical.

  For example, the following method, exported by 4D in Text mode...

```
C_STRING(10;$1)

If (Count parameters=0)
   <>pr_Document:=New process("List_Documents";64*1024;"List of Documents";"Open";*)
   SHOW PROCESS(<>pr_Document)
   BRING TO FRONT(<>pr_Document)
Else
   If ($1="Open")
      MENU BAR(2)
      message_for_document:=""
      page_document:=Load list("od_Page_Document")

      COPY ARRAY(<>Document_types;Document_types)

      ALL RECORDS([od_Document])

      INPUT FORM([od_Document];"Input")
      OUTPUT FORM([od_Document];"List")
      $ref:=Open form window([od_Document];"Input")

      MODIFY SELECTION([od_Document];*)
      CLOSE WINDOW
      <>pr_Document:=0
   End if

   If (Is a list(page_document))
      CLEAR LIST(page_document;*)
   End if
End case
```

... will appear as follows in a text editor:

```
C_STRING(10;$1)

If (Count parameters=0)
<>pr_Document:=New process("List_Documents";64*1024;"List of Documents";"Open";*)
SHOW PROCESS(<>pr_Document)
BRING TO FRONT(<>pr_Document)
Else

If ($1="Open")
MENU BAR(2)
message_for_document:=""
page_document:=Load list("od_Page_Document")

COPY ARRAY(<>Document_types;Document_types)

ALL RECORDS([od_Document])

INPUT FORM([od_Document];"Input")
OUTPUT FORM([od_Document];"List")
$ref:=Open form window([od_Document];"Input")

MODIFY SELECTION([od_Document];*)
CLOSE WINDOW
<>pr_Document:=0
End if

If (Is a list(page_document))
CLEAR LIST(page_document;*)
End if
End case
```

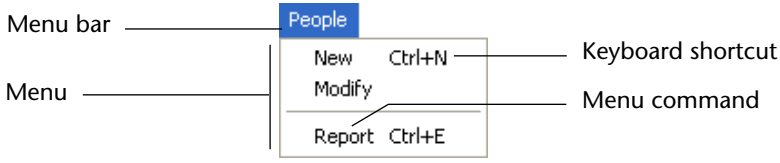
9 Creating Custom Menus

You can create custom menus for your databases and custom applications. Because pull-down menus are a standard feature of any desktop application, their addition will make your databases easier to use and will make them feel more familiar to users. When you create custom menus, you can also create custom toolbars. With custom menus and toolbars, your databases will perform more like "stand-alone" applications.

A custom application must contain at least one menu bar with one menu. By default, when you create a new database, 4th Dimension creates a custom menu bar so that you can access the Custom Menus environment. For detailed information about creating custom applications, refer to the *4th Dimension Language Reference* manual.

Designing Menus

In general, menus provide menu commands that the user chooses to perform database tasks: modifying records, searching for records, printing reports, and so on. The figure below shows an example of a custom menu.



A menu bar is a group of menus that can be displayed on a screen together. Each menu on a menu bar can have numerous menu commands in it, including separator lines that divide the menu commands into groups and keyboard shortcuts. When the user chooses a menu command, it calls a project method or a standard action that performs an operation.

You can have many separate menu bars for each database. For example, you can use one menu bar that contains menus for standard database operations and another that becomes active only for reporting. One menu bar may contain a menu with menu commands for entering records. The menu bar appearing with the input form may contain the same menu, but the menu commands are disabled because the user doesn't need them during data entry.

You can also use the Menu Bar editor to create custom toolbars. To do so, you associate an icon with a menu command. The icon appears in the 4th Dimension toolbar and the text of the menu command is used as the icon's Tip.

When you use the same menu for more than one menu bar, you can take advantage of the concept of "instances" of a menu to simplify the process of managing menus. For complete information about this method of managing menus, see the section "Working With Instances of a Menu" on page 601.

When you design menus, keep the following two rules in mind:

- **Use menus for functions that are suited to menus** Menu commands should perform tasks such as adding a record, searching for records, or printing a report.
- **Group menu commands by function** For example, all menu commands that print reports should be in the same menu. For another example, you might have all operations for a certain table in one menu.

You use the Menu Bar editor to create menus. You can perform the following operations in the Menu Bar editor:

- Create and rename menu bars,
- Specify styles for menu commands,
- Specify keyboard shortcuts for menu commands,
- Add dividing lines to menus,
- Associate a project method or a standard action with a menu command,
- Assign password groups to menu commands,
- Enable or disable menu commands,
- View sample menus while you are creating the menu bar,

- Paste in a custom graphic associated with each menu bar that will be displayed as a splash screen,
- Create a custom toolbar that uses an icon for each menu command,
- Create a connected menu,
- Specify that a new process start when a menu command is chosen.

Each of these tasks is explained in this chapter.

Creating Menus

4th Dimension allows you to create entire menu bars. A menu bar is the collection of menus that appears at the top of your application window. The menu bar displays the menu titles and the menus pull down to display the menu commands. Every menu command should be associated with a project method or a standard action.

4D Server Object locking occurs when two or more users attempt to modify the same menu bar at the same time. If a user is modifying a menu in a menu bar, the menu is locked. Other users can modify different menus in that menu bar, but they cannot modify the same menu. In addition, if a user is modifying any aspect of a menu bar, other users cannot add any new menus to the menu bar.

Basic Steps for Creating Menus

The following are the basic steps for creating custom menus:

1 **Create one or more menu bars.**
See the next section, "Creating a Menu Bar" on page 593.

2 **Create the menus that will pull down from the menu bar.**
See the section "Adding Menus" on page 595.

3 **Add menu commands to each menu in the menu bar.**
See the section "Adding Menu Commands" on page 596.

4 **Assign a project method or a standard action to each menu command.**
When the user chooses that menu command, 4th Dimension executes the method or standard action associated with it. See the section "Assigning Methods to Menu Commands" on page 598.

5 **Write the project methods that perform the menu commands.**
For more information about using the Method editor, see the section "Using the Method Editor" on page 548. Refer to the *4th Dimension Language Reference* manual for detailed information about 4th Dimension's programming language.

6 **Make any enhancements you want such as special font styles, separator lines, keyboard shortcuts, a toolbar icon, and so on.**
See the section "Enhancing Menus" on page 603.

7 **Assign password access groups to menu commands (optional).**
See the section "Assigning a Group To Database Objects" on page 631.

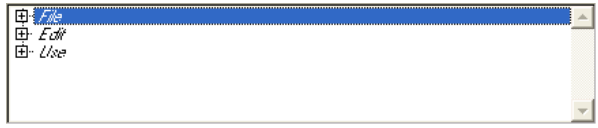
8 **Specify that a new process be started when a menu command executes (optional).**
See the section "Assigning Methods to Menu Commands" on page 598.

Default menu bars

When you create a new database, 4th Dimension automatically creates a default menu bar (Menu Bar #1) including standard menus and a menu for navigating between the environments.

This allows the user to access the Custom Menus environment as soon as the database is created. Menu Bar #1 is called automatically when the **Custom Menus** command is chosen in the **Use** menu.

The default menu bar includes three menus: **File**, **Edit** and **Use**.



Note This default menu bar is also generated when you add a new menu bar.

- **File**: This menu only includes the **Quit** command. The *Quit* standard action is associated with the command, which actually causes the application to quit.
- **Edit** (standard): The **Edit** menu is standard and completely modifiable. Editing functions such as copy, paste, etc. are defined using standard actions.
  In databases created with former versions of 4th Dimension, the **Old Edit menu mechanism** option is available. For more information, refer to the paragraph "Former Edit menu mechanism," page 595.

- **Use**: The **Use** menu contains, by default, the **Design**, **User** and **Custom Menus** commands. This menu allows access to the different 4D environments directly from the Custom Menus mode.

  You can modify this menu bar as desired or create additional ones.

Creating a Menu Bar

This section describes the process of creating a custom menu bar.

► To create a menu bar:

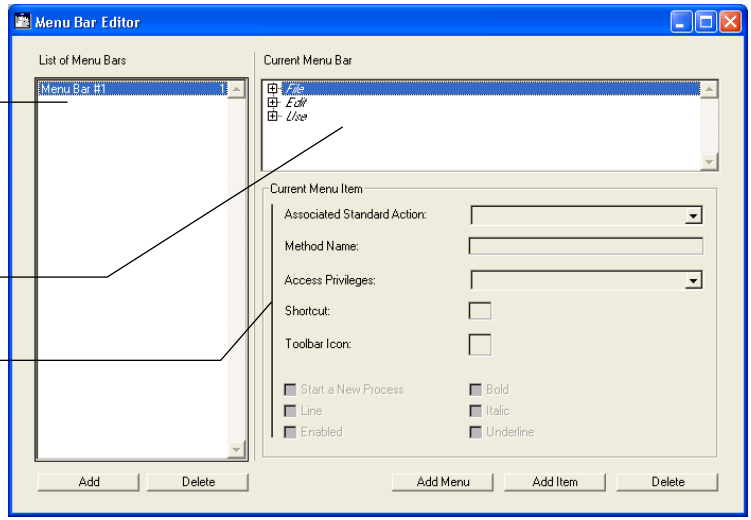
1 **Choose Menu Bar Editor from the Tools menu.**

4th Dimension displays the Menu Bar editor. By default, Menu Bar #1 appears in the panel on the left— as well as any other menu bars that may already have been created.

The List of Menu Bars shows the existing menu bars

The Current Menu Bar area shows the menus belonging to the selected menu bar

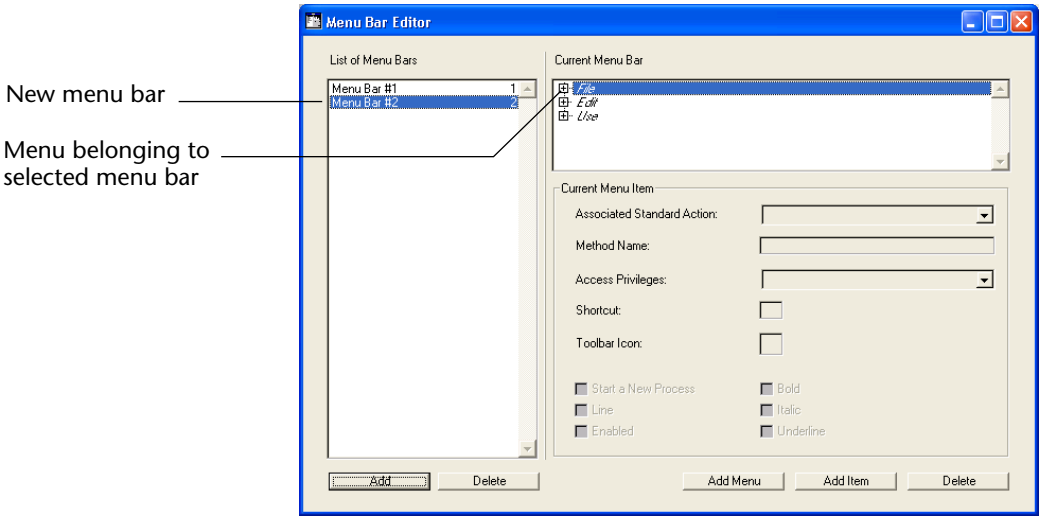
The Current Menu Item area shows the properties of the selected menu command



4th Dimension assigns menu bar numbers sequentially — Menu Bar #1 appears first. You can rename menu bars but you cannot change their numbers.

2 **Click the Add button under the "List of Menu Bars" area.**

A new menu bar appears in the list.



3 Double-click on the name of the menu bar in order to make it modifiable and enter a custom name (optional).

Note You can also make the menu bar name modifiable by clicking on it while holding down the **Ctrl** (Windows) or **Command** (MacOS) key.

A custom name can facilitate the identification of menu bars in the different dialog boxes of the Design environment and in the language commands. A menu bar name may contain up to 31 characters and must be unique.

Note It is not possible to rename a menu bar installed by a component, no matter what its status (public, protected or private).

At this point, you can begin modifying menu bars or adding menus, commands to the menus, etc.

The Menu Bar editor displays menu information in the following three lists:

- **List of Menu Bars** displays the name of each menu bar.
- **Current Menu Bar list** displays the name of each menu in the menu bar currently selected in the List of Menu Bars. The Current Menu Bar list is a hierarchical list; each menu can be expanded to display the menu commands belonging to the menu.

- **Current Menu Item area** displays the properties of the selected menu command.

  Notice that **File**, **Edit**, and **Use** menus as well as their menu commands appear italicized in the Current Menu Bar list. The italics indicate that the text for the menu title or command is being retrieved from a string resource. If you press the **Ctrl** key (Windows) or the **Command** key (MacOS) while clicking the **File** menu, the string resource number ":79, 1" appears. For more information about this point, refer to the paragraph "Using a resource reference for a label," page 596.

Former Edit menu mechanism

In databases created with previous versions of 4th Dimension, an additional option, **Old Edit menu mechanism**, is available. This option is checked by default for existing menu bars. It enables you to keep the former operation of the **Edit** menu. When this option is checked, 4th Dimension automatically adds an **Edit** menu to the menu bar. In this case, the menu is managed by the system and cannot be modified.

Adding Menus

You can add menus to either a new or an existing menu bar. You can add a menu to a menu bar in the following two ways:

- Create a new menu,
- Create a connected menu.

  When you create a connected menu, you are attaching an exact duplicate of an existing menu to a menu bar. This duplicate is called an *instance* of that menu. For information about creating an instance of a menu, see the section "Creating Connected Menus" on page 602.

  Before you can create a connected menu, the first instance of that menu must be created in the Menu Bar editor window.

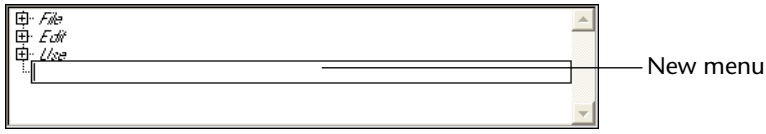
  You can create a menu in the following two ways:

- Append a new menu to the end of the current list of menus,
- Insert a new menu anywhere in the current list of menus.

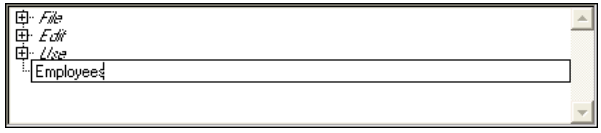
  You don't have to create the menus in the order that they will eventually appear. You can rearrange menus after you've created them using drag and drop. For more information, see the section "Rearranging Menus and Menu Commands" on page 598.

► To add a menu:

1 **Choose Add Menu from the Menu menu.**
OR
Click the Add Menu button.
4th Dimension adds a new menu to the Current Menu Bar list so that you can enter the menu title.



2 **Type the name of the new menu in the entry area.**
The maximum length for a menu title is 15 characters. Additional characters are ignored.



Using a resource reference for a label

You can use a STR# resource instead of text. This can facilitate the translation of an application. For example, if you enter ":2000,3", the text contained in the third line of the STR# 2000 resource will be displayed as the menu label. Changing the contents of this STR# resource by programming using the language commands of 4th Dimension or a resource editor (under MacOS) will change the name of the menu accordingly the next time it is displayed. Similarly, if you enter, for instance, "<>vlang,3", the text contained in the third line of the STR# resource whose number is found in the <>*vlang* interprocess variable will be displayed as the menu label. You just have to change the contents of this variable in order for the label of the menu to be modified the next time it is displayed.

3 **Repeat steps 1 and 2 to add more menus (if desired).**

Adding Menu Commands

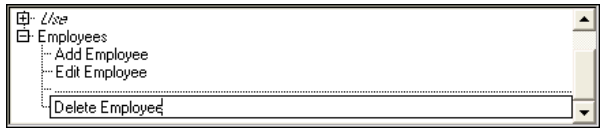
For each menu in the menu bar, you must create the menu commands that appear when the menus are pulled down.

► To add a menu command:

1 **Select the menu that you want to provide with a menu command.**

2 **Click Add Item.**
OR
Choose Add Item from the Menu menu.
4th Dimension adds a new item to the Current Menu Bar list.

3 **Type the name of the new menu command.**
The figure below shows **Delete Employee** being added as a menu command.



Notice that the third menu command appears blank. This menu command is a separator line. For information about creating a separator line, see the section "Adding Separator Lines" on page 606.

The figure below shows the menu commands and the separator line in the sample menu.

4 **Repeat steps 1 through 3 to add more menu commands to the Items list.**

When you create a menu, 4th Dimension displays it to the right of the current menu bar in order for you to have a preview of the way it will appear in your application. You can scroll this "sample" menu in the same way as a 4th Dimension menu. When you create menu items, they appear in the "sample" menu.

Using Control Characters in Menu Labels

You can define the properties of the menu commands by using control characters (metacharacters) directly in the menu command labels. For instance, you can assign the keyboard shortcut **Ctrl+G** (Windows) or **Command+G** (MacOS) for a menu command by placing the "/G" characters in the label of the menu item label. Control characters do not appear in the menu command labels. You therefore need to make sure you are not using them when they are not used as control characters.

The control characters are the following:

- "("
- "<"

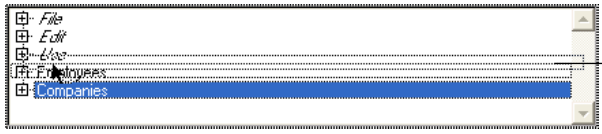
- "!"
- "^"
- "/"

For more information on the use of these characters, refer to the description of the APPEND MENU ITEM in the *4th Dimension Language Reference* manual.

Rearranging Menus and Menu Commands

After you create the menus for a menu bar and the menu commands for a menu, you can reorder them using drag and drop. To insert a menu command at a different place in the order, simply drag the menu command to the new location. To move a menu, simply drag a menu to another location in the list of menus.

The following illustration shows a menu being dragged:



Assigning Methods to Menu Commands

To enable a menu command to perform its function, you must assign either a project method or a standard action to it.

These methods or standard actions perform the functions indicated by the menu commands. For example, the **Monthly Report** menu command can call a project method that prepares a monthly report from a table containing financial data. The **Cut** menu command can call the *Cut* standard action in order to move the selection to the clipboard and erase it from the window in the foreground. When a menu command is chosen, 4th Dimension executes the standard action or project method that is assigned to it.

The choice between associating a standard action or a project method with a menu command depends on the type of result desired. In principle, it is preferable to choose a standard action whenever possible since they implement optimized mechanisms.

Note Standard actions are not compatible with the mechanism that converts menu bars into lists of URLs, which is provided by the Web server of 4th Dimension in contextual mode. For more information on this point, refer to the *Language Reference* manual of 4th Dimension.

You can also assign both a standard action and a project method to a menu item. In this case, the standard action is never executed; however, 4th Dimension uses this action to enable/disable the menu item according to the context. When a menu item is disabled, the associated project method cannot be executed.

You create the project methods in the Method editor. You can create them either before or after you assign them to the menu command. When you have assigned a method to a menu command in the Menu Bar editor, you can open this method by simply selecting the menu command in the Current Menu Bar area and pressing **Ctrl+P** (Windows) or **Command+P** (MacOS).

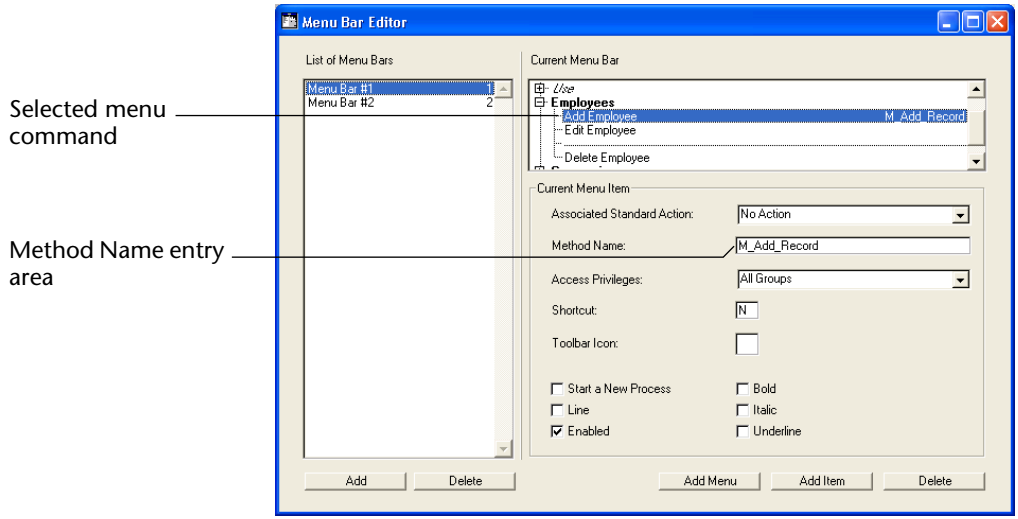
▶ To assign a method to a menu command:

1 **Create or select the menu command.**

4th Dimension highlights the selected menu command. The Current Menu Item area changes to show the properties of the selected menu command.

2 **Drag a method name from the Methods page of the Explorer to the Method Name area in the Menu Bar editor or type the name of the method in the Method Name area.**

If you typed the method name, press Tab or click outside the entry area to save the method name.



You can add a method name before you write the method.

When you finish entering the name, it appears in the Current Menu Bar area.

Note If you change the name of a method that is used in a menu, you must update the method name here in the Menu Bar editor.

3 Click the Start a New Process check box (optional).

If you click the **Start a New Process** check box, a new process is created when the menu command is chosen.

Normally, a method attached to a menu command executes within the current process unless you explicitly call New process in your code. The **Start a New Process** check box makes it easier to start a new process.

If you click the **Start a New Process** check box, 4th Dimension will create a new process when the menu command is chosen. In the Process list, 4th Dimension assigns the new process a default name using the format *M_ProcessNumber*. The names of processes started from a menu are created by combining the prefix "M_" with the process number.

For more information about processes, see the chapter "Managing Processes," page 671.

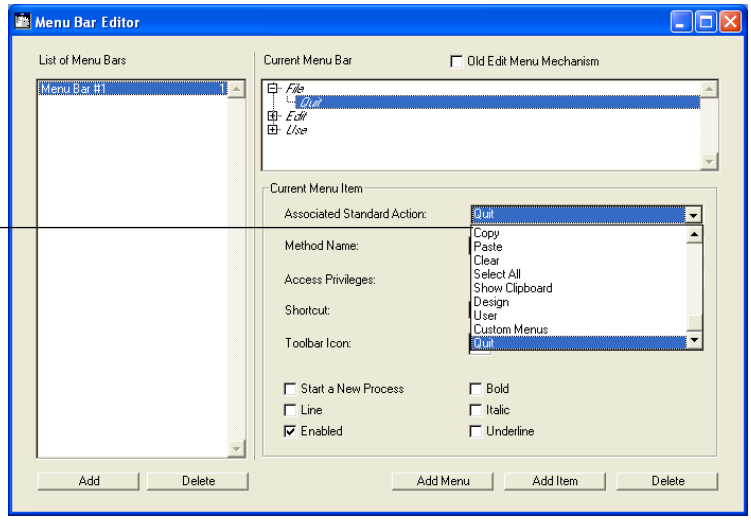
Note When the menu command is called via a URL from the Web server home page in contextual mode, the server receives the special 4DMETHOD URL if the **Start a New Process** option is checked. In this case, you must be sure that the method has the "Available through 4DACTION" attribute (for more information, refer to the "Connection Security" section in the *4th Dimension Language Reference* manual.

► To associate a standard action with a menu command:

1 Create or select the menu command.

4th Dimension highlights the selected menu command. The Current Menu Item area then displays the properties of this menu command.

2 **Choose the action you want to assign to it in the "Associated Standard Action" menu.**



Selection of a standard action for the selected menu command

The list of standard actions proposed for menus is similar to the one for buttons (accessible using the palette of button properties in the 4th Dimension Form Wizard). Most of the actions can, in fact, be used in both contexts. However, the *Edit Subrecord*, *Delete Subrecord*, *Add Subrecord* and *Automatic splitter* actions cannot be associated with menu commands: they do not appear in the pop-up selection menu. For a description of each standard action, refer to the paragraph "Standard Button Actions," page 424.

Note for MacOS X Under MacOS X, the Custom Menus command associated with the **Quit** action is automatically placed in the application system menu, when the database is running in this environment. This mechanism simplifies the management of the **Quit** command under MacOS X.

Working With Instances of a Menu

When you write a custom application, you will usually find that you reuse certain menus in several places in your application. The same menu may be attached to several menu bars.

If you create a menu from scratch each time you use it in a menu bar, you must manage each menu separately, on a menu bar by menu bar basis. If you want to change a menu (disable or enable a menu com-

mand, place a check mark next to a menu command, and so on), you must change it every place that it occurs.

If you take advantage of the concept of "instances" of a menu by creating connected menus, managing menus becomes much easier. With connected menus, it is possible to update a menu *wherever it occurs* in a single step.

Creating Connected Menus

When you create a menu using the **Add Menu** command in the **Menu** menu or the **Add Menu** button, you create the first instance of that menu.

When you want to reuse the menu in another menu bar, you connect the menu to that menu bar. Connecting the menu attaches another instance of the menu to the menu bar.

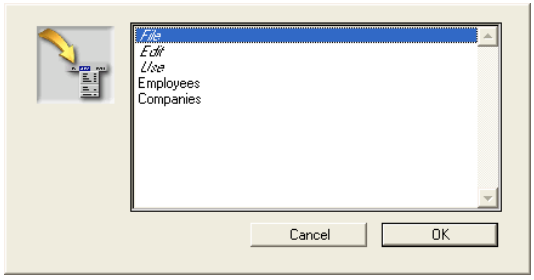
► To connect a menu to a menu bar:

1 **Make sure that the menu bar to which you want to connect is the currently selected menu bar in the Menu Bar editor.**

2 **Select the menu that you want to appear below the connected menu.**

3 **Choose Connect Menu from the Menu menu.**

The Select Menu dialog box appears.



4 **Select the name of the menu you want to connect.**

5 **Click the OK button.**

This connects another instance of the menu to the menu bar. The connected menu appears directly above the currently selected menu in the Menu Bar editor.

Modifying Connected Menus

The most important thing to remember about connected menus is that all instances of a menu refer to a single menu description. If you modify any instance of a menu, all instances of the menu reflect the change.

Modifications include the following:

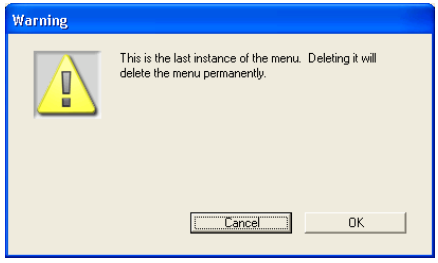
- Changes to the name of the menu, menu commands, or associated methods or standard actions,
- Enhancements to the menus, such as adding separator lines, disabling or enabling menu commands, changing fonts or styles, and assigning keyboard shortcuts,
- Modifications produced with the 4th Dimension language, such as disabling a menu command by using the DISABLE ITEM command.

Deleting Connected Menus

Although connected menus refer to a single menu description, there is no main instance of a menu to which other instances refer.

When you delete a menu, you are only deleting one occurrence of the menu. The menu is not removed from the database — it is only deleted from a single menu bar — unless you delete the only occurrence of this menu.

If you attempt to delete the only occurrence of a menu in your database, a confirmation dialog box appears to remind you that deleting it will remove the menu from your database permanently.



For complete information about deleting menus, see the paragraph "Deleting Menus and Menu Commands," page 609.

Enhancing Menus

You can change the font style of menu commands, add separator lines between groups of menu commands, assign a keyboard shortcut for a menu command, and enable or disable menu commands. You can also create a custom toolbar by assigning icons to menu commands.

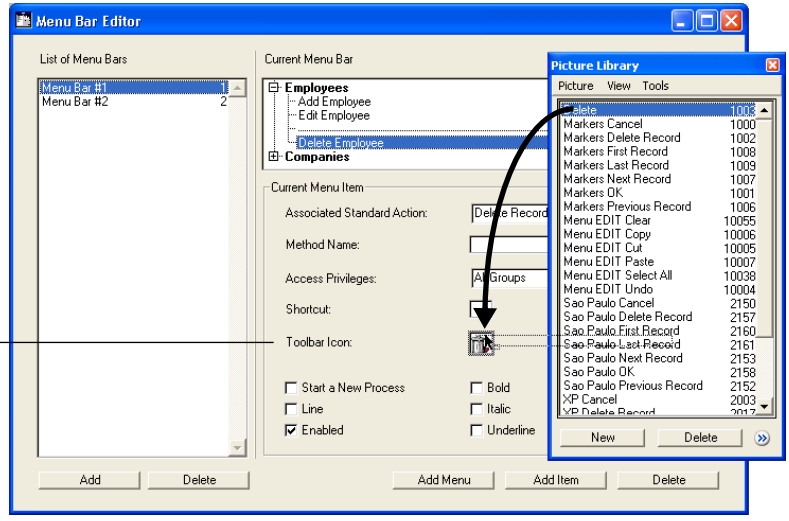
Adding an Icon to a Menu Command

You can associate an icon with a menu command. When you do so, the icon is used as a button in the toolbar that is displayed whenever the menu bar is displayed.

► To add an icon:

1 **Highlight the menu command to which you want to associate an icon.**
The Current Menu Item area changes to show the properties of the selected menu command.

2 **Open the Picture Library, select a graphic, and drag the picture from the Picture Library to the In Toolbar area.**
4th Dimension displays the icon in the area.



Icon from Picture Library added to menu item

For preview purposes, the icon appears immediately to the right of the standard 4th Dimension toolbar in the Menu Bar editor.

Preview of the icon

When the menu bar is used in the Custom Menus environment, 4th Dimension displays the icon in the toolbar and automatically uses the text of the menu command as the Tip for the toolbar button.

Changing Font Styles

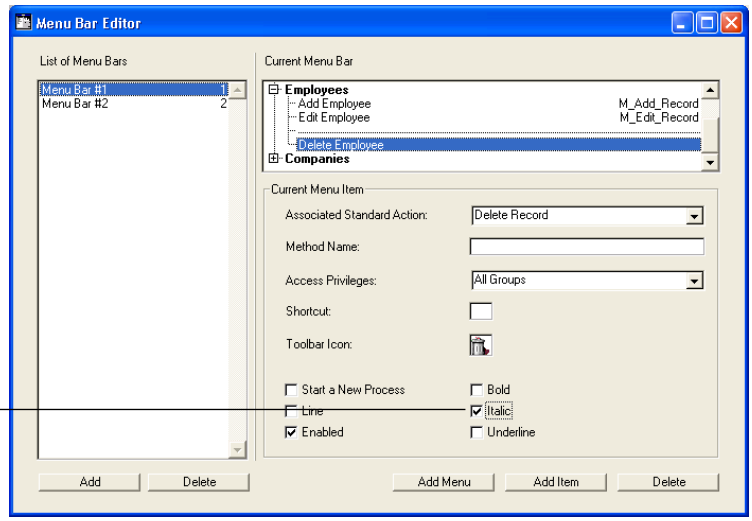
4th Dimension lets you customize menus by applying different font styles to the menu commands.

You can customize your menus with the Bold, Italic or Underline styles.

Be cautious when applying font styles to your menus — too many styles will be distracting to the user and give a cluttered look to your application.

To apply a style, select the menu command you want to modify and then choose the style from the check boxes.

The following illustration shows an italic style being applied to a menu command:



The sample menu appears like this:

Enabling and Disabling Menu Commands

You can specify whether a menu command will appear enabled or disabled. An enabled menu command can be chosen by the user; a disabled menu command is dimmed and cannot be chosen. Unless you specify otherwise, 4th Dimension automatically enables each menu command you add to a custom menu[1].

Note You can also enable or disable menu commands using methods.

1. If you add a separator line, the Menu Bar editor automatically disables the line.

► To enable or disable a menu command:

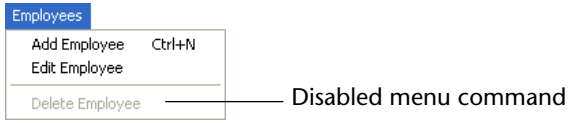
1 **Select the menu command you want to enable or disable.**

2 **To enable the menu command, select the Enabled check box.**

To disable the menu command, uncheck the **Enabled** check box.

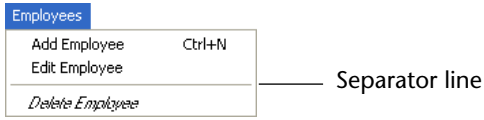
When the **Enabled** check box is unchecked, the menu command appears dimmed, signifying that it cannot be chosen.

A disabled command appears as follows in the sample menu.



Adding Separator Lines

Groups of menu commands in a menu can be separated by a separator line. This convention is useful for grouping associated menu commands by function.



You add a separator line by creating a menu command. Instead of entering the menu command's text in the Current Menu Bar area, you simply select the Line property. Instead of text, the line appears in the Current Menu Bar area.

► To add a separator line:

1 **Select the menu you want to change in the Current Menu Bar list.**

The menu commands for that menu appear in the Items list.

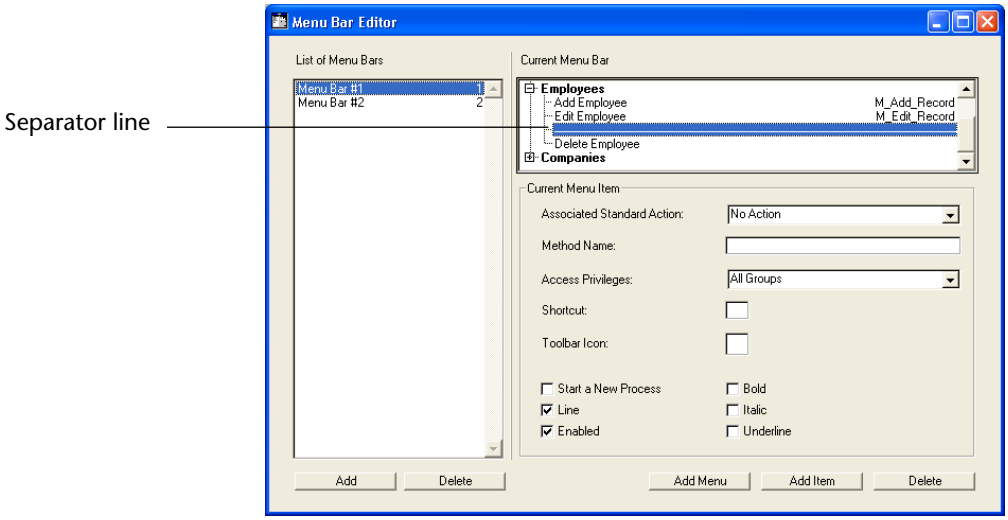
2 **Choose Add Item from the Menu menu.**
OR
Click Add Item.

4th Dimension creates a new menu command.

3 **Click the Line check box property in the Current Item properties area[1].**

1. Under MacOS, if you use the dash "-" as the first character of a menu item, it will appear as a separator line. This is especially useful when using the APPEND MENU ITEM command.

The line appears in the Current Menu Bar area.



Note When the **Line** option is checked, the other properties have no effect.

Assigning Keyboard Shortcuts

You can add keyboard shortcuts to any menu command. If a menu command has one of these keyboard shortcuts, users will see a symbol for it next to the menu command. For example, "**Ctrl+C**" appears next to the **Copy** menu command on the **Edit** menu. This means you can copy a selection by holding down the **Ctrl** key and pressing **C**.

Note It is advisable to keep the default keyboard shortcuts associated with standard actions.

4th Dimension lets you assign keyboard shortcuts to the menu commands in your custom menus. You can use any alphanumeric keys in combination with the **Ctrl** key (**Command** key on MacOS) as a keyboard shortcut, except for the keys reserved by standard menu commands that appear in the **Edit** and **File** menus and the keys reserved for 4th Dimension menu commands.

These reserved key combinations are listed in the following table.

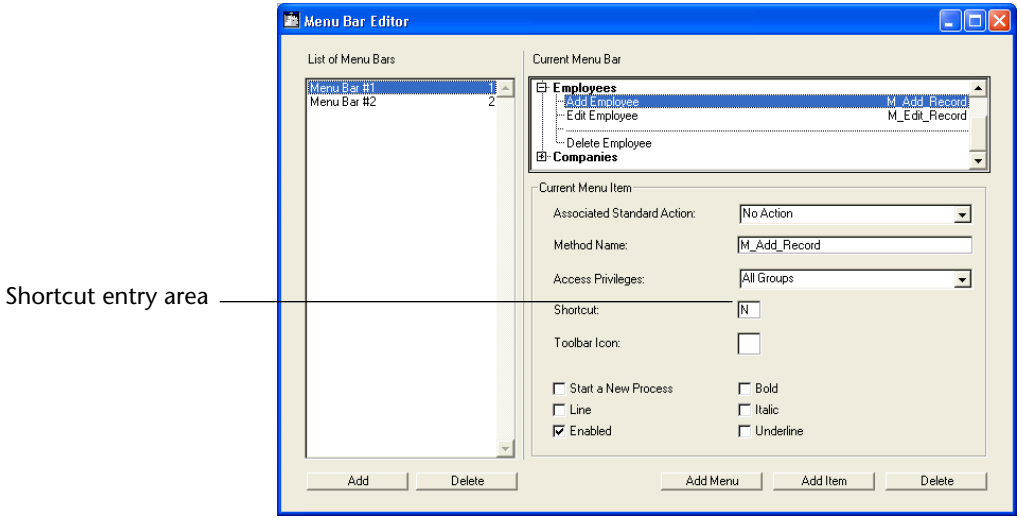
Key[1]	Operation
Ctrl+C	Copy
Ctrl+Q	Quit
Ctrl+V	Paste
Ctrl+X	Cut
Ctrl+Z	Undo
Ctrl+. (period)	Stop action
Ctrl+W	Flushes records to disk in User or Custom Menus environments

1. Under MacOS, use Command instead of Ctrl.

► To assign a keyboard shortcut:

1 **Select the menu command to which you want to assign a keyboard shortcut.**

2 **Type the alphanumeric character that you want to associate with the menu command in the "Shortcut" entry area.**



Tip If the text of a menu item is followed by the slash and a letter, the Menu Bar editor will automatically use the letter as the keyboard shortcut. For example, if you enter "Add Employee/N" as the text of a menu item, the Menu Bar editor will automatically use Ctrl+N as the keyboard shortcut.

The user can choose the menu command by holding down the **Ctrl** key (Windows) or **Command** key (MacOS) and pressing the assigned key.

Note On Windows, keyboard shortcuts using the Alt key are automatically handled by the system, not by 4th Dimension.

You can pull down the sample menu to see how the keyboard shortcut appears in the menu (optional).

Note An active object can also have a keyboard shortcut. If Ctrl key assignments conflict, the active object takes precedence. For information on assigning keyboard shortcuts to active objects, see the section "Assigning a Keyboard Shortcut" on page 418.

Deleting Menus and Menu Commands

You can delete an instance of a menu at any time. A deleted menu no longer appears on the menu bar. You might not want a particular menu on a certain menu bar. You might delete a menu after you have placed its menu commands on other menus. Or, you might delete menus that you have placed on different menu bars.

For information about deleting connected menus, see the section "Deleting Connected Menus" on page 603.

► To delete a menu:

1 **Select the menu you want to delete in the Current Menu Bar list.**
When you select a menu, the names of menu commands assigned to the menu appear in the Items list.

2 **Click Delete.**
OR
Choose Delete from the Menu menu.
4th Dimension removes the menu from the menu bar. The deleted menu will no longer appear in the application's menu bar.

You can delete a menu command at any time. You might delete a menu command that is out of date. You might delete a menu command after you have placed it on another menu. Or, you might delete a menu command that has been replaced by another menu command.

► To delete a menu command:

1 **Select the menu command you want to delete.**

2 **Choose Delete from the Menu menu.**
OR
Click Delete in the Current Menu Bar area.
4th Dimension removes the menu command from the Current Menu Bar list.

Previewing Menu bars and Adding a Splash Screen

As you develop your custom application, 4th Dimension lets you view the custom menus and menu bars as they will appear in the application.

You can also embellish each menu bar with a "splash screen," a custom graphic displayed under the menu bar when it appears. A splash screen can include a company logo or other design elements.

Before you can add a splash screen, you need to create the graphic using a graphics application, by scanning, or by some other means.

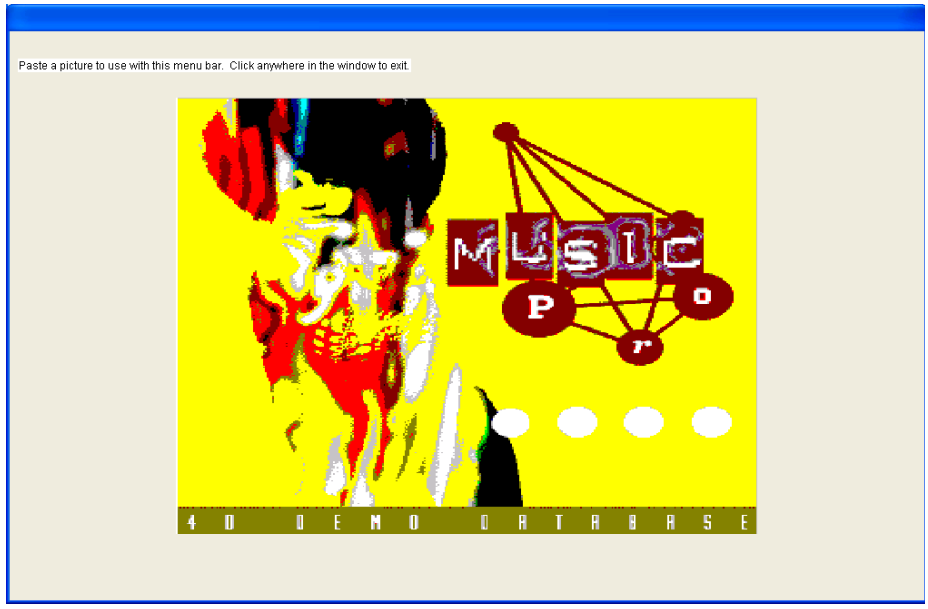
► To preview the menu bar and add a splash screen:

1 **Choose Show Custom Menus from the Menu menu.**
4th Dimension displays the menu bar as it will appear in the custom application and prompts you to copy a picture to accompany the menu bar.

2 **Open the menus listed on the menu bar to preview the contents of the menus.**

3 **If you are going to add a splash screen, copy the graphic to the Clipboard and paste it into the Splash Screen area.**
4th Dimension centers the graphic image beneath the menu bar.

The following illustration shows an example of a custom splash screen.



4 **When you are satisfied with your splash screen, click anywhere in the screen to exit the preview.**

The splash screen will appear whenever this menu bar is used in the custom application when no form is displayed.

Note You can modify the default size of this window in your 4th Dimension databases using the 4D Customizer Plus utility ("Windows" resource). For more information, refer to the 4D Customizer Plus user manual.

Menus and Custom Applications

Menu bars provide the major interface for custom applications. For each custom application, you must create at least one menu bar with at least one menu. See the *4th Dimension Language Reference* manual for more information about creating custom applications.

You can create menu bars for the Custom Menus environment regardless of whether you are creating a custom application or simply creating menus for use in the Custom Menus environment. By default, Menu Bar #1 is the menu bar displayed in the Custom Menus environment. You can change which menu bar is displayed using a method.

If you define a menu command without assigning it a method, choosing that menu command exits the Custom Menus environment. If you are using the application with 4D Runtime, leaving the Custom Menus environment takes you to the Program Manager (Windows) or Finder (MacOS).

If you are using the full 4th Dimension application, a password access system can be set up to control where each user is placed after leaving the Custom Menus environment. Indeed, you can define an access group for the User Environment in the Preferences dialog box. Users who do not belong the access group set for the User Environment will not be able to access it from the Custom menus mode by either selecting a menu command, or by pressing the **Alt + F4** keys (Windows) or **Option + F** (MacOS). When users that do not have the adequate access privileges attempt to switch to the User Environment, 4th Dimension quits.

Note The Designer and the Administrator will always have access to the User Environment, even if they do not belong to the User Environment access group. Users that have access to the Design environment will always have access to the User Environment, even if they do not belong to the User Environment access group. For more information, refer to the chapter "Managing Password Access," page 613.

10 Managing Password Access

If more than one person uses a database, you may want to control access to the database or provide different capabilities and interfaces to different users. If you are designing applications for use in a multi-user environment or the World-Wide Web, it may be essential that you provide security for sensitive data. You can provide this security by assigning passwords to users and creating access groups that have different levels of access to information in the database or database operations.

This chapter provides information about 4th Dimension's Password Access editor. You use the Password Access editor to:

- Specify the users of a database,
- Provide users with passwords,
- Create groups of users with different levels of access to the database,
- Nest groups of users within other groups to create a hierarchy of users,
- Define a default user,
- Specify the group which owns the objects each user creates,
- Assign a startup method for each user,
- Specify a group owner.

After you create access groups, you can manage access to:

- The Design Environment
- The User Environment
- Table properties
- Record operations
- Forms
- Methods
- Menu commands
- Plug-ins

In addition to providing security for your databases, the access system also maintains a user history — the Password Access editor can tell you how many times a user has used the database and the date of the most recent use.

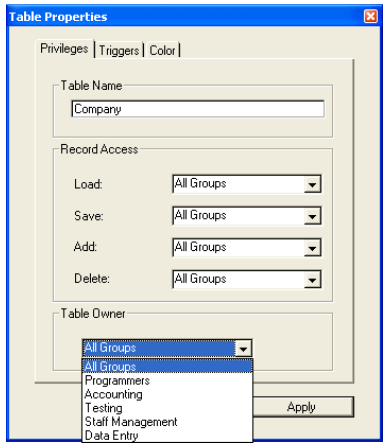
4D Server Object locking occurs when two or more users attempt to modify the password access system at the same time. Only one user can use the Password Access editor at a time.

Access System Overview

4th Dimension's password access system is based on users and groups. You create users and assign passwords, put users in groups, and assign each group access rights to appropriate parts of the database.

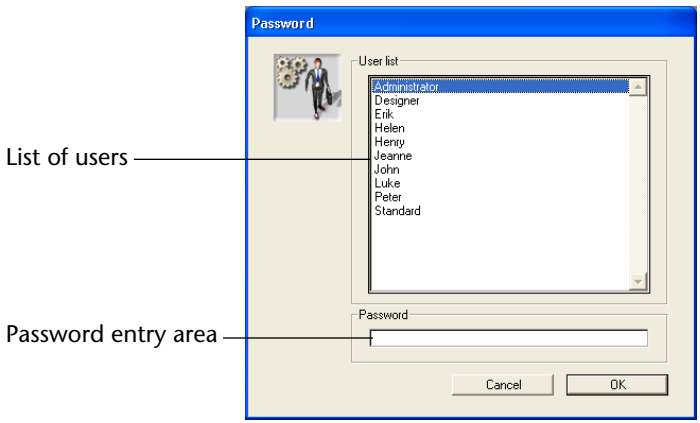
Groups can be assigned access privileges to operations on records in the table and to the table definition.

The figure below shows table owner privileges being assigned to a group:



To open the database, a user either selects or types his or her user name and types his or her password[1]. Then, depending on which groups the user belongs to and to which parts of the database the groups have been assigned, the user can operate the parts of the database that were specified by the access system.

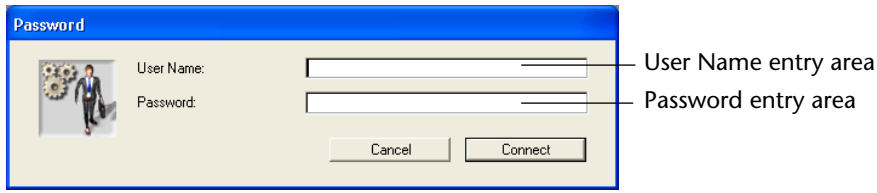
The **Access Management** page ("Database" theme) of the Preferences dialog box lets you set the access mode of the database (see the paragraph "Access Management Page," page 132). By default, the following Enter Password dialog box is displayed.



1. Unless the Default User option is used (for more information, refer to "Defining a Default User" on page 622).

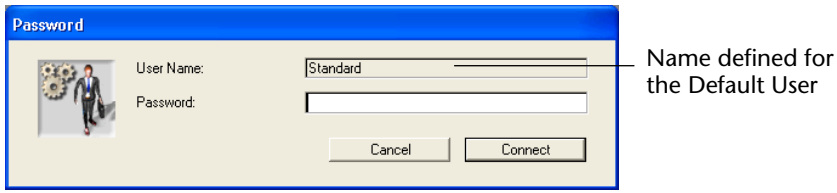
In this dialog box, the user selects his or her name from the list of users and types his or her password in the password entry box.

If you deselect **Display User List in Password Dialog Box** in the Preferences dialog box, the Enter Password dialog box shown below will be displayed.



In this dialog box, the user must type both his or her name and password.

If you have defined a Default User in the Preferences dialog box and have assigned it a password, the following dialog box is displayed:



Users only have to enter the password. If no password is assigned to the Default User, the dialog box is not displayed. Each user, in this case, has the same privileges and restrictions as defined for the Default User.

4D Server After logging in to the database using any of the above dialog boxes, the user may choose to save the location of the server database (the pathname) and (optionally) the password used to log in to the database. The next time that the user double-clicks the 4D Client application icon, the database is automatically run and, if the user has saved his or her password, the user is automatically logged in to the database. For more information, refer to the *4D Server Reference* manual.

The user operates the database in a normal fashion. When the user attempts to use a form, menu command, method, or table that the group is not permitted to use, 4th Dimension displays an error message of the type "Your password does not allow you to use this form."

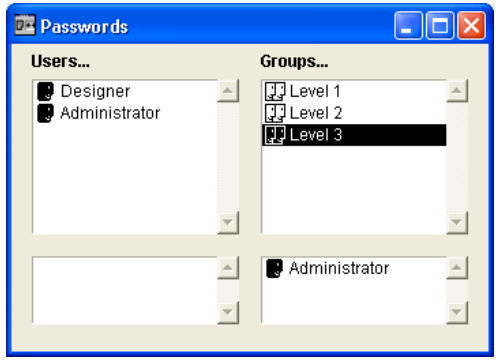
Note If an *ON ERR CALL* method is installed, the error message for methods and tables is not displayed. Refer to the *4th Dimension Language Reference* manual for more information.

An Access Hierarchy Scheme

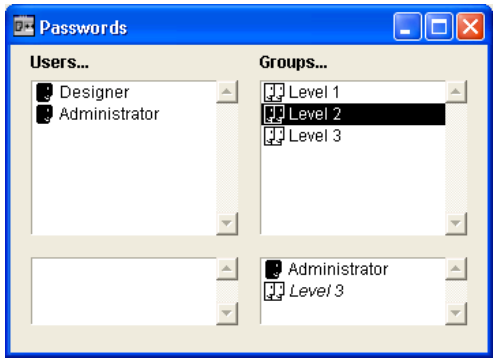
The best way to ensure the security of your database and provide users with different levels of access is to use an access hierarchy scheme. Users can be assigned to appropriate groups and groups can be nested to create a hierarchy of access rights. This section discusses some approaches to such a scheme.

In this example, a user is assigned to one of three groups depending on the user's responsibility. Users assigned to group Level 1 are responsible for data entry. Users assigned to group Level 2 are responsible for maintaining the data, including updating records and deleting outdated records. Users assigned to group Level 3 are responsible for analyzing the data, including performing searches and printing analytical reports.

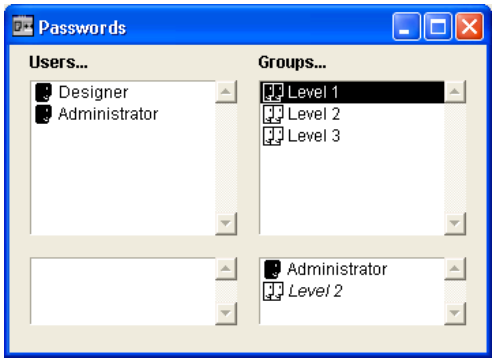
The groups are then nested so that privileges are correctly distributed when the groups are assigned. Level 3 contains only "high-level" users.



Level 2 contains data maintenance users as well as Level 3 users so that the users in Level 3 have the privileges of Level 2 as well.



Level 1 contains data entry users as well as Level 2 users so that users who belong to Level 2 and Level 3 enjoy the privileges of Level 1 as well.



You can decide which group to assign access privileges to based on responsibility. If you assign group Level 1 to an input form, for example, it means that everyone can use this input form. If you assign group Level 2 to the form, it is restricted to members of Level 2 and Level 3. If you assign group Level 3, only members of Level 3 can use the form.

Such a hierarchical system makes it easy to remember to which group to assign a new user. You only have to assign each user to one group and you use the hierarchy of groups to determine access.

As part of designing your access system, another consideration to keep in mind is the level at which a user should be restricted. You can think of each of the parts of a database to which access can be controlled — methods, forms, tables and table operations — as being part of a natural hierarchy. For example, if only a table is restricted, a user may try to display a form only to be restricted at the table level.

Your access scheme should restrict access at the highest appropriate level, usually at the form level.

The Designer and the Administrator

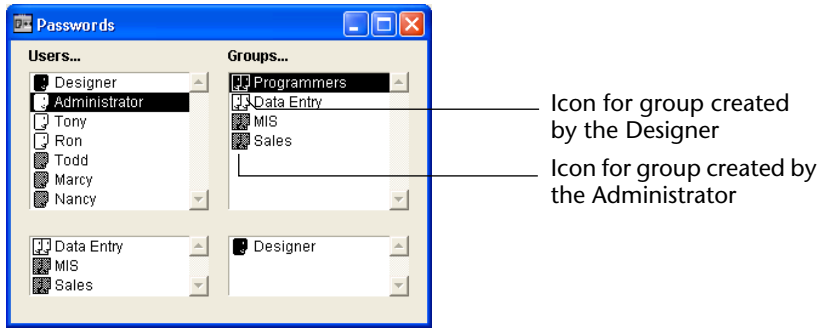
4th Dimension provides users with certain standard access privileges to the two environments and certain powers within each environment. Once a password access system has been initiated, these standard privileges take effect.

The most powerful user is named Designer. The Designer has control over the design of the database. The Designer can create users and groups, assign access privileges to groups, and use the both the User and Design environments. No aspect of the database is closed to the Designer, except for users or groups created by other users.

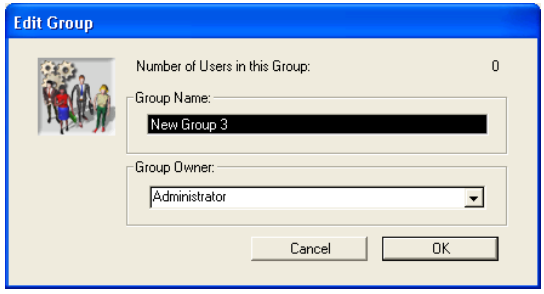
After the Designer, the next most powerful user is the Administrator, who is usually given the task of managing the password access system. When the Password Access editor is first opened, both the Designer and Administrator appear in the list of users. At this point, the Administrator is just a regular user with no special access privileges. To be able to use the password access system, the Administrator must be given the ability to enter the Password Access editor. For information about allowing the Administrator to enter the Password Access editor, see the section "Administrator and Group Owner Access" on page 638. The Administrator is the only user with the ability to save and load groups. For information about saving and loading groups as the Administrator, see the section "Saving and Loading Groups" on page 630.

The Administrator's access to other parts of the database is limited by group membership — the Administrator must be part of one or more groups to have access privileges in the database. The Administrator is placed in every new group, but you can remove the Administrator's name from any group.

You can distinguish between users and groups created by the Designer and Administrator by the color of their icons. The icons for users and groups created by the Designer are white while those created by the Administrator are grey. The icons for the Designer and Administrator themselves are both black.



The figure below shows a group being created by the Administrator. The default name of the group is Group 1.



The group owner can change the default name at any time.

The Designer and the Administrator can each create 16,000 groups and 16,000 users.

Group Owners

You can designate an owner for each group. Usually, the owner is the Administrator, but you can designate any group member as the owner.

The group owner can be given the ability to add and remove users from any group he or she owns. The users to be added must already exist. Group owners cannot create users, or create or change user passwords. Group owners cannot add or remove other groups.

Like the Administrator, the group owner must be given the ability to enter the Password Access editor, normally reserved to the Designer of the database. For information about allowing a Group owner to enter the Password Access editor, see the section "Administrator and Group Owner Access" on page 638.

Giving Users Design Environment Access

Normally, the Designer of a database is the only user who has access to the Design environment. However, if you have a number of users who need to be able to modify the database design, you can place those users in a group and give the group access to the Design environment.

You give a group access to the Design environment by selecting the group from the **Structure Access** drop-down list on the "Access Management" page of the Preferences dialog box. For more information, see the paragraph "Access Management Page," page 132.

All other users are ordinary users. When a user opens the database, it opens in the Custom Menus environment or in the User environment (if the access is authorized). The access of a user is limited by group membership.

Giving Access to the User Environment

You can define a group's access to the User environment. This option allows you to control and protect the access to the User environment when a user is in the Custom Menus environment. A user that is not part of the group that has access to the User environment will not be able to access the User environment from the Custom Menus environment, either by using a menu command or by using the standard shortcut (**Option+f** under MacOS, **Alt+F4** or close box under Windows). If a user tries to access the User mode without having the appropriate privileges, 4th Dimension automatically quits.

The Designer and Administrator always have access to the User environment even if they are not explicitly part of the group that has access to the User environment.

A user that has access to the Design environment always has access to the User environment, even if the user is not explicitly part of the group that has access to the User environment.

You give a group access to the Design environment by selecting the group from the **Structure Access** drop-down list on the "Access Management" page of the Preferences dialog box. For more information, see the paragraph "Access Management Page," page 132.

Defining a Default User

You can define a Default User to use your database. When this option is active, users that open or connect to the database are no longer required to enter a name. Moreover, if you have not associated a password with the Default User, the Password dialog box does not appear and the database opens directly.

Once logged in as the Default User, each user has the access privileges and restrictions defined for the Default User.

This option simplifies access to the database while maintaining a complete data control system.

► To define a Default User:

1 **In the Design environment, create a user (the name you choose is not important) in the Password editor.**

You can associate a password with the user, but it is not mandatory. For more information refer to "Assigning Users and Groups" on page 624.

2 **Using the Design environment editors, choose the access privileges and restrictions for this user.**

For more information refer to "Assigning a Group To Database Objects" on page 631.

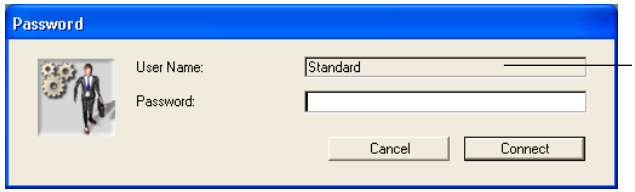
3 **In the Preferences window, go to the Access Management page ("Database" theme).**

4 **Choose your user in the Default User drop-down list (located at the bottom of the window).**

5 **Accept the dialog box.**

The access to the database is now no longer customized.

- If you have associated a password with the Default User, a dialog box appears when the database is opened and the Default User's password must be entered:



- If you haven't associated a password with the Default User, the dialog box doesn't appear.

Redisplaying the Password Dialog Box

You can force 4th Dimension to display the standard Password dialog box to, for example, connect to the database as the Designer or Administrator.

▶ To redisplay the Password dialog box when the Default User mode is active:

1 **Open the database while holding down the Shift key.**

A Password dialog box appears allowing you to enter a name and password.

Initiating the System

You initiate the 4th Dimension password access system by assigning a password to the Designer.

Until you give the Designer a password, 4th Dimension allows anyone to use any part of the database.

When a password is assigned to the Designer, all the access privileges you have assigned to tables, forms, menus, and methods take effect. In order to open the database, users must enter a password.

Important ***Do not forget the Designer's password! If you do, you will be unable to open the database in the Design environment.***

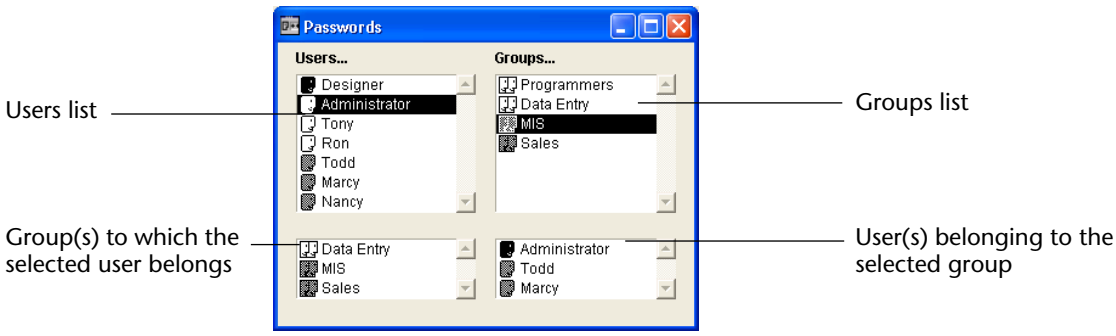
To disable the access system, you just need to remove the Designer password.

Assigning Users and Groups

You use the Password Access editor to create access groups and users and to assign passwords to users.

1 **To open the Password Access editor, choose Passwords from the Tools menu.**

4th Dimension displays the Password Access editor window:



The Password Access editor window displays four scrollable areas. The upper Users list displays user names. The users designated Designer and Administrator appear at the top of the list. The lower Users list displays the names of any groups a selected user belongs to.

The upper Groups list displays the names of access groups. The lower Groups list displays the names of users who belong to a selected group.

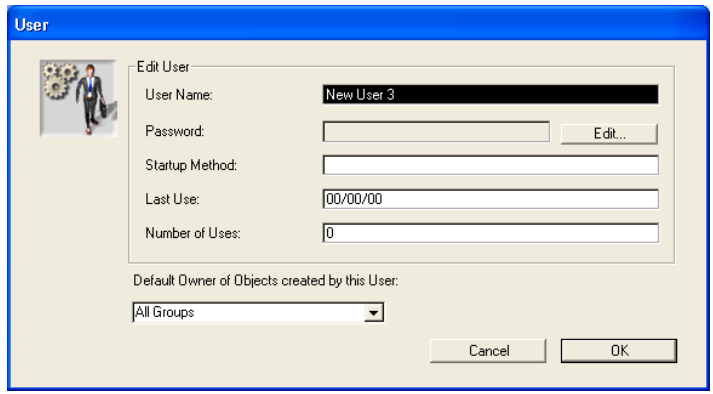
When the Password Access editor is active, the **Passwords** menu is available. You use the menu commands on this menu to add users and groups.

Adding Users

To add a user and assign user information:

1 **Choose New User from the Passwords menu.**

4th Dimension displays the User dialog box.

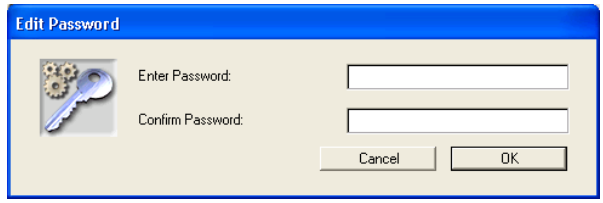


The dialog box provides areas for entering a user name, password, startup method, and the group which owns the objects that the user creates. In addition, the Designer or Administrator can view information about an individual's use of a database or database application.

2 **Type a new user name.**

3 **Click the Edit button.**

The Edit Password dialog box is displayed:



4 **Type the password in the Enter Password entry area and enter it again in the Confirm Password area.**

You can use up to 15 alphanumeric characters for a password. The Password Access editor is case sensitive — the user must enter the password exactly as it is entered in the Edit Password dialog box. For example, if you define a user's password as "HolyCow," the user must enter it with a capital H and capital C, or 4th Dimension will not accept the password.

After you specify a user password, that password is not available for view the next time you open the dialog box. Asterisks display in place of the password characters.

5 **Click OK to validate the password.**

If the two password entries are different, 4th Dimension plays a Beep and the dialog box is not closed.

6 **Type a startup method to be executed when the user opens the database (optional).**

7 **Choose a group from the Default Owner of Objects Created by This User drop-down list.**

This group owns any objects (tables, forms, methods, and so on) that the user creates. For instance, you might specify that the MIS group owns the objects created by each user in the MIS group. If a user from another group attempts to modify a form created by a member of the MIS group, a message appears stating that the user does not have privileges to edit the form.

8 **Click the OK button to save the user information.**

► To change user names and passwords:

1 **From the Password Access editor, select the user name in the list of users and double-click the user name.**
OR
Select the user's name and choose Edit User from the Passwords menu.
4th Dimension displays the Edit User dialog box.

2 **Make the modifications you want in the dialog box.**

3 **Click the OK button to save the changes you made to the user profile.**

Users added by the Designer cannot be removed. To remove users added by the Administrator, see the section "Saving and Loading Groups" on page 630.

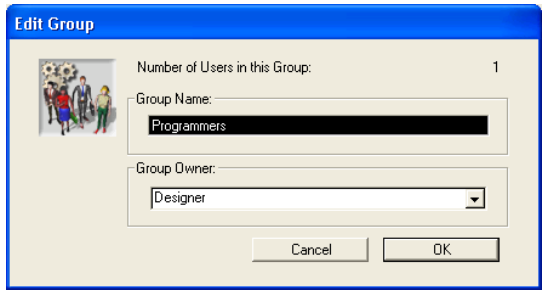
Creating Access Groups

When you create a group, you can designate a group owner from among the users.

► To create a group:

1 **Choose New Group from the Passwords menu.**

4th Dimension displays the Edit Group dialog box.



2 **Enter a group name in the "Group Name" box.**
The group name can be up to 15 characters in length.

3 **Click on the "Group Owner" box and select a user from the drop-down list of users.**
The group owner can add users to and remove users from the group.

4 **Click the OK button to add the group to the access system.**
The name of the new group appears in the group list.

► To change access groups:

1 **From the Password Access editor, double-click the group name in the list of groups.**
OR
Click on the group name and choose Edit Group from the Passwords menu.
4th Dimension displays the Edit Group dialog box.

2 **Enter a new group name or specify a new group owner in the dialog box.**

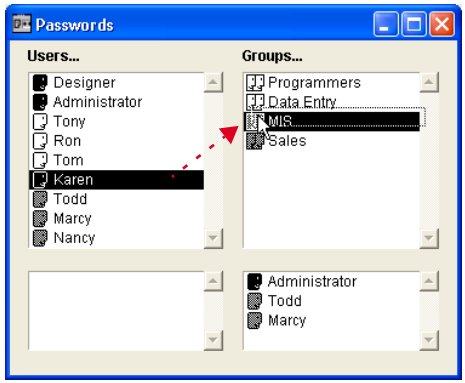
3 **Click the OK button to save the changes you made to the group.**

Assigning Users to Groups

You can assign users to any group and you can assign a user to several access groups. You are not required to assign a user to a group.

To assign a user to a group, drag the user name from the Users list over the name of the desired group in the Groups list and release the mouse button.

The figure below shows a user name being added to a group.



The user name then appears in the lower Groups list of users assigned to the selected access group. The group name appears in the lower Users list of groups assigned to a selected user. The assigned user now has all the privileges that you provide that group.

Click any group name to see a list of users who belong to that group.

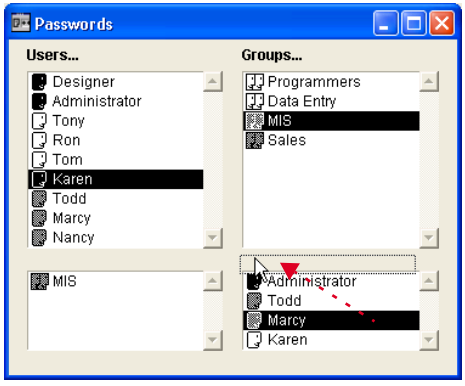
Removing Users from Groups

► To remove a user from an access group:

1 **Click the group name in the upper Groups list to view the names of users assigned to the group.**

2 **Drag the user name from the lower Groups list to the Users list and release the mouse button.**

The figure below shows a user being removed from a group.



The user no longer has the privileges that you provide that group.

Nesting Groups

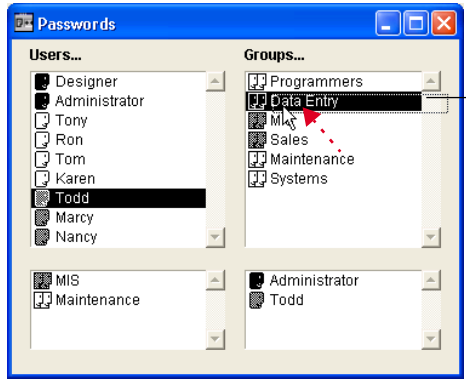
To create a user hierarchy, you nest groups, placing one group within another. The users of the nested group obtain the privileges of both groups. For example, if you nested the Executive group inside the Data Entry group, users assigned to the Executive group would automatically get the privileges of Data Entry in addition to the privileges accorded to Executive. However, users inside Data Entry are denied access to the privileges of the Executive group — they have only the privileges assigned to Data Entry.

For further explanation of how a hierarchical access system works, see the section "An Access Hierarchy Scheme" on page 617.

► To assign a group to another group:

■ **Drag the group name from the list of groups over the name of the group whose privileges you want it to assume and release the mouse button.**

The figure below shows a group being added to another group.



The nested group name appears in italics in the list of users assigned to the selected group. In addition to its assigned privileges, the nested group inherits all the privileges of the group it has been placed in.

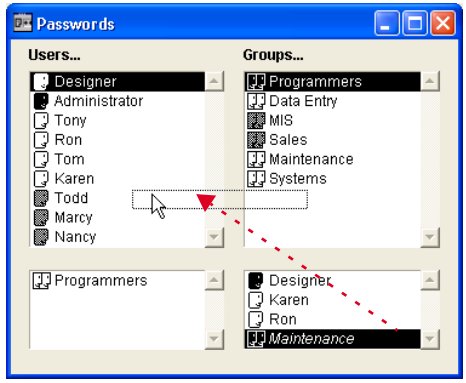
Removing Nested Groups

► To remove a group from another group:

1 **Select the group name in the upper Groups list to display the names of users and groups assigned to the group.**

2 **Select the group name in the lower Groups list, drag the name to the Users list, and release the mouse button.**

The following illustration shows the Maintenance group being removed from the Programmers group.



The group name disappears from the list of users for the selected group.

Saving and Loading Groups

4th Dimension allows the Administrator to save and load any groups that he or she has created or modified. When groups are saved, everything about the current users and groups are saved. Later, when loaded again, the original users and groups are installed.

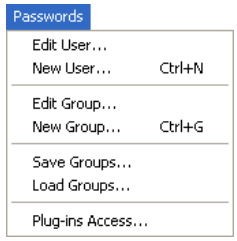
The ability to save groups means that the Administrator can save the access system of a database and transfer it to a modified version of the same database or to a new database. This is extremely useful for restoring the access system after the Designer has provided a new, updated version of the database. Because the groups can be reloaded, users of the database do not have to learn a new access system. All the user names, passwords, startup method names, groups, group owners, and group memberships are preserved.

Note The Designer cannot save or load groups.

► To save the current groups:

1 Enter the database as the Administrator.

2 Choose Save Groups from the Passwords menu.



4th Dimension displays a create-file dialog box so that you can name and save the group.

► To load groups:

1 Enter the database as the Administrator.

2 Choose Load Groups from the Passwords menu.

4th Dimension displays an open-file dialog box so that you can open the Groups file.

Assigning a Group To Database Objects

After you define users and access groups, you can assign groups to the following objects:

- Table properties and operations,
- Forms,
- Methods,
- Menu commands,
- Plug-ins.

You may assign only one group to each object. For this reason, it is important to design the access groups so that more powerful users belong to all the groups below them in the access hierarchy.

For a discussion of how to organize users and access groups, see the section "An Access Hierarchy Scheme" on page 617.

Assigning Access To Record Operations

You can assign different groups to each of the four record operations. You can thus specify which groups can load, save, add, or delete records from a table. These privileges can be very sensitive for some databases.

The following are the record operations that groups can be assigned to:

- **Load** This operation allows users to open and view records. It does not provide the right to modify records, create new records, or delete records.
- **Save** This operation allows users to save modified records. It does not provide the right to load records, add new records, or delete records.
- **Add** This operation allows users to create new records. It does not provide the right to load records, modify existing records, or delete them.
- **Delete** This operation allows users to delete records. It does not provide the right to load records, modify existing records, or add new records.

  You must provide Load privileges for any group that has Save or Delete privileges.

  You can allow some users the right to add records without being able to load and view any other records. You can allow others the right to modify records, but not add any new ones.

  For each table in a database, you can assign a group to be the owner of the table definition. Users in this group can modify specifications for the table, including the groups given access to record operations.

► To assign access privileges to record operations for a table and to the table's definition:

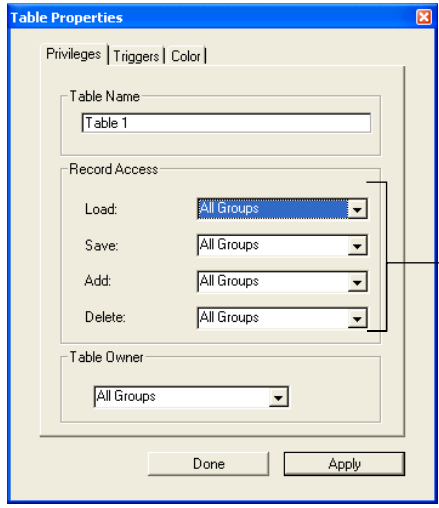
1 **In the Structure editor window, select the table image of the table whose access privileges you want to edit[1].**

  The selected table image is surrounded by a marquee.

2 **Double-click the table title.**
  OR
  Choose Table Properties from the Structure menu.
  OR
  Click the table image using the right mouse button (Windows) or click the table image while pressing the Ctrl key (MacOS) and choose Table Properties from the contextual menu.

1. You can also double-click the name of the table on the Tables page of the Explorer to view the table properties for the table.

4th Dimension displays the Privileges page of the Table Properties window.



3 **Choose a group for each database operation from the drop-down lists in the "Record Access" area.**

4 **Choose a group from the "Owner" drop-down list in the Table Access area.**

Users in this group can modify the definition of the table in the Table Properties window.

5 **When you have finished, click the Apply button.**

Assigning a Group to a Form

When you assign a group access privileges to a form, only users belonging to that group can use that form for data entry. When you assign a group owner privileges to a form, only users belonging to that group can modify that form in the Design environment.

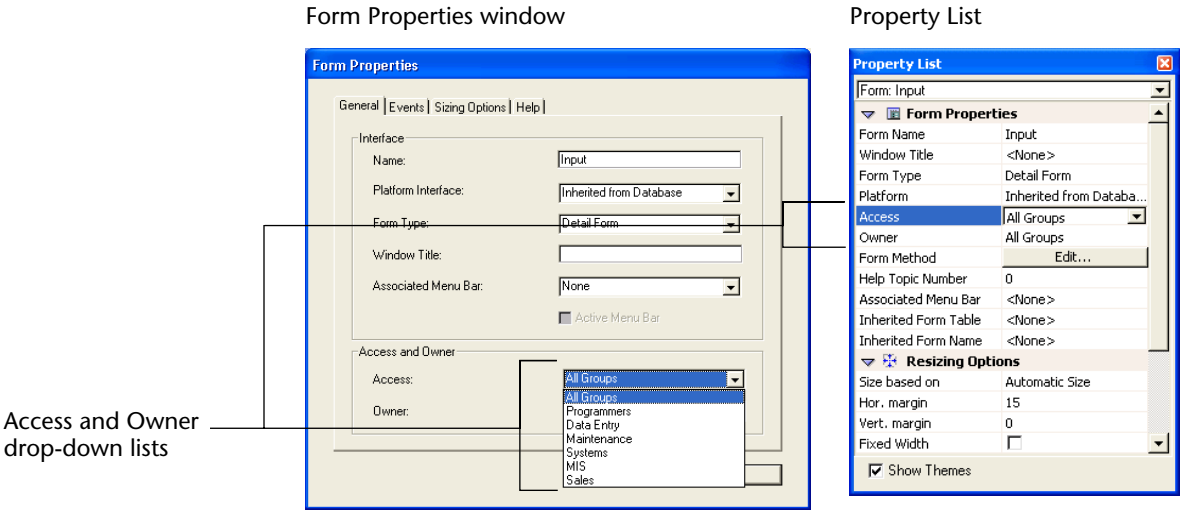
► To assign access and owner privileges:

1 **Open the form in the Form editor.**

For directions on opening a form, see the section "Opening a Form in the Form Editor" on page 288.

2 **Choose Form Properties from the Form menu.**
OR
Display the form properties in the Property List.

For more information about this choice, refer to "Displaying and Setting Form and Object Properties" on page 282.



3 **Choose a group from the "Access" drop-down list.**

If you do not choose a group, all groups retain access privileges to the form (the default).

4 **Choose a group from the "Owner" drop-down list.**

If you do not choose a group, all groups retain owner privileges to the form (the default).

Assigning a Group to a Project Method

When you assign a group access to a project method, only users belonging to that group can use that method. When you assign a group ownership of a project method, only users belonging to that group can modify that method in the Design environment.

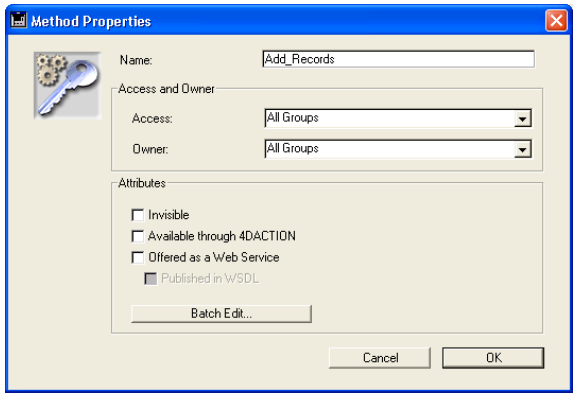
► To assign access and owner privileges:

1 **Open the method in the Method editor.**

For directions on opening a project method, see the section "Creating or Opening Methods" on page 534.

2 **Choose Method Properties from the Method menu.**

The Method Properties dialog box appears.



3 **Choose the group from the "Access" drop-down list.**

If you do not choose a group, all groups retain access privileges to the method (the default).

4 **Choose a group from the "Owner" drop-down list.**

If you do not choose a group, all groups retain owner privileges to the method (the default).

5 **Click the OK button.**

Assigning a Group to a Menu Command

You can assign an access group to a menu command so that only users in that group can use the menu command in the User or Custom Menus environments.

► To assign an access group to a menu command:

1 **Choose Menu Bar Editor from the Tools menu.**

The Menu Bar editor appears.

2 **Select a menu bar.**

The Current Menu Bars area shows the menus belonging to this menu bar.

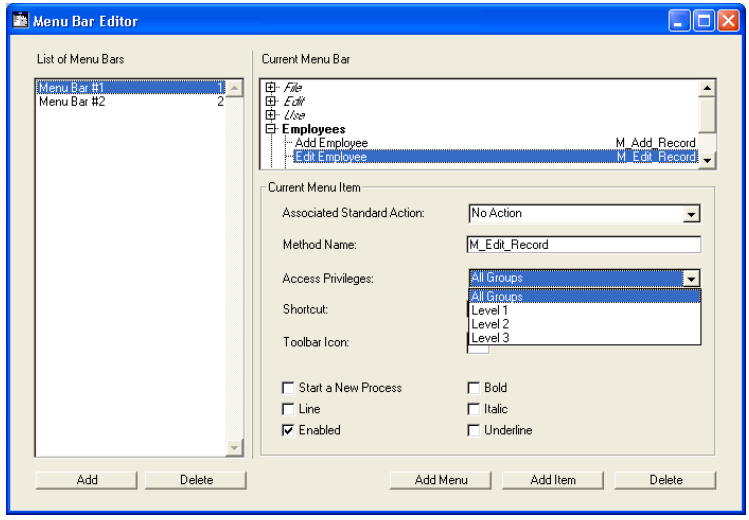
3 **Expand a menu in the Menus list.**

The menu commands and methods for this menu appear.

4 **Select the menu command for which you want to specify an access group.**

5 **Select the group from the "Access Privileges" drop-down list.**

The figure below shows All Groups being given access privileges to the **Edit Employee** menu command.



Assigning a Group to a Plug-in or to the 4D Client Web server

You can assign a group privileges to any plug-ins installed in the database. This includes all the 4th Dimension plug-ins and any third-party plug-ins.

You can also restrict the use of the 4D Client Web server via the Access to Plug-ins dialog box.

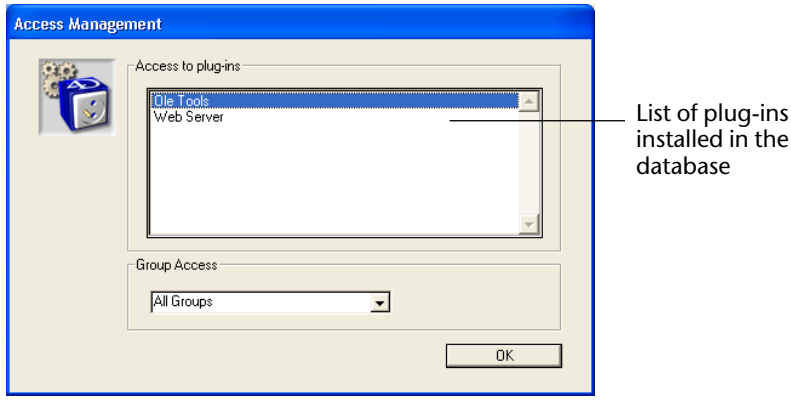
► To assign an access group to a plug-in or the 4D Client Web server:

1 **Choose Passwords from the Tools menu.**

The Password Access editor appears.

2 **Choose Plug-Ins Access from the Passwords menu.**

The Access to Plug-ins dialog box appears with a list of the plug-ins installed in the database.



The **Web Server** element lets you control the possibility of Web publication for each 4D Client. Client Web licenses are considered as plug-in licenses by 4D Server. Therefore, in the same way as for plug-ins, you can restrict the right to use Web Server licenses to a specific group of users. For more information, refer to the "Web server configuration and connection management" section in the *4th Dimension Language Reference* manual.

3 **Choose the plug-in for which you want to assign a group.**

4 **Choose the group from the "Group Access" drop-down list.**

5 **Click the OK button.**

System Maintenance

Once a password access system is in place, occasional maintenance of the system is necessary. Users must be added, groups need new members, and passwords need to be changed. The Designer has access to the Design environment and can make any necessary modifications using the Password Access editor.

The Administrator and the Designer can also view the usage history of each user as necessary for maintenance.

Administrator and Group Owner Access

The Administrator does not necessarily have access to the Design environment. However, if the Designer creates a project method that contains the EDIT ACCESS command, the Administrator and group owners can have limited power to control users and groups.

The EDIT ACCESS command can be included in a method that is attached to a custom menu or can be executed by choosing **Execute Method** from the **Special** menu in the User environment. If the method is executed by a user who is not the Administrator or another group owner, it has no effect.

When the method is executed, the result depends on whether the user is the Administrator or a group owner.

If the Administrator executes the method that contains the EDIT ACCESS command, 4th Dimension displays the Password Access editor. The Administrator can use the Password Access editor to create users and groups; edit any users or groups he or she created, including changing user passwords; and add or remove users from any groups he or she created.

The Administrator cannot assign groups to forms, tables or table operations, menu commands, methods, or plug-ins. Only the Designer can assign these access groups.

If a group owner who is not the Administrator executes the method that contains the EDIT ACCESS command, 4th Dimension displays the Password Access editor, but displays only the groups that the group owner owns. The group owner can add or remove users from the groups. The group owner cannot create users, edit user information, or add groups. The menu commands for adding and editing users and groups are dimmed.

Viewing Usage

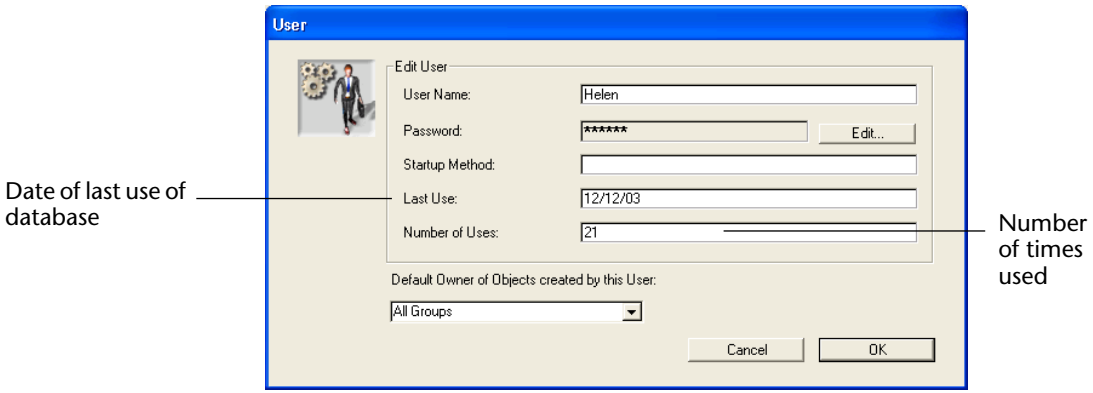
The Edit User dialog box contains the date of the user's last use of the database and the total number of uses. The Administrator or Designer can view this information by opening the Edit User dialog box for any user.

► To open the Edit User dialog box:

1 **Choose Passwords from the Tools menu.**
OR
Execute the method that contains the EDIT ACCESS command.
4th Dimension displays the Password Access editor.

2 Select the user name you want to see from the Users list and choose Edit User from the Passwords menu.

The Edit User dialog box displays the date of the user's last use of the database as well as the number of times the user has opened the database.



11 Creating Lists

This chapter tells you how to create and use lists. A list is a set of possible values. You can use a list to do the following:

- Provide the user choices from which to select an entry for a field or enterable object,
- Restrict the valid entries to those in the list,
- Exclude the entries in the list from being entered.

4th Dimension lets you associate a small icon with each item in a list or hierarchical list. Where appropriate, the small icon is displayed to the left of the item. For example, you can display the small icons in hierarchical lists.

When a list is used as a choice list for a field or enterable object, the user can simply select from the list instead of typing the entry. For example, you may want to create a choice list for entering job titles in a personnel database.

You can also use lists to provide restrictions on data entry. One list may provide the required values for a field, excluding all others. Another list may provide the excluded values for a field, preventing any value in the list from being entered.

You can also create hierarchical lists. A hierarchical list associates a sub-list with each element of the list.

Your lists can offer up to 8,000 choices in a single database and each choice can contain up to 255 characters.

For information about adding a choice list to a field as a field attribute, see the section "Choices and Help" on page 188. For information about using lists with data entry controls, see the section "Using Choice Lists" on page 374.

Lists are often used in methods. For example, a list is a convenient place to store the elements of an array. An array stores a list of values in memory. You can use lists to store the elements of pop-up menus, hierarchical lists, combo boxes, tab controls, and other multi-valued interface objects. You transfer the contents of the list to the interface object using a method or by assigning the list to the object in the Object Properties window.

You create 4th Dimension lists with the List editor. You use the List editor to do the following:

- Create lists and hierarchical lists,
- Add items to lists,
- Associate small icons with list items,
- Delete lists,
- Delete items from a list,
- Sort items in a list,
- Make a choice list user-modifiable
- Make hierarchical list items editable or a tab control active,
- Specify the spacing between hierarchical list elements,
- Insert hierarchical lists or hierarchical pop-up menus in forms.

4D Server Object locking occurs when two or more users attempt to modify the same list at the same time. If a user is modifying a list in the Design environment, the list is locked. Other users cannot modify the list, the list name, or any of the items in the list, until the first user frees the list by closing it.

Designing Lists for Data Entry

One use of lists is to provide the user with a list of values from which to choose during data entry. The following are some considerations about lists that stem from this purpose:

- You can make a list available for every form or for selected forms.
- You can restrict the possible entries to those in the list or you can allow the user to type additional entries.

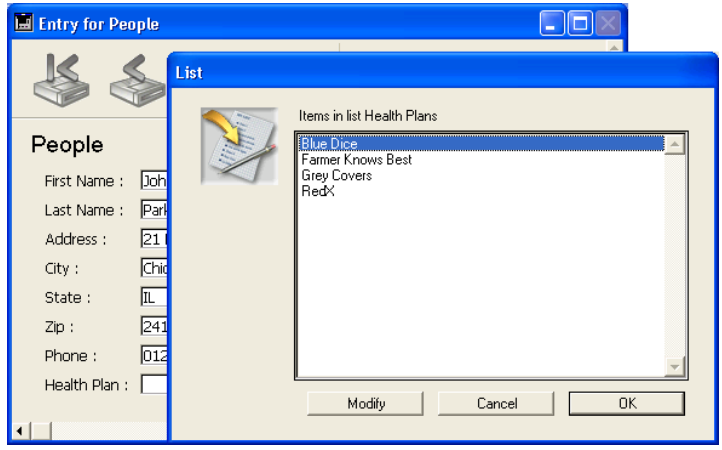
- You can allow the user to modify the list or you can prevent the user from modifying the list.

  You can attach a list to a field as one of the field properties. Attaching a list to a field causes the list to appear whenever that field is selected during data entry or whenever an output form is used in the Enter in List mode. The user can select an entry from the list. If the list is sorted, the list automatically scrolls as the user types characters at the keyboard. For instance, if the user types "N," the list scrolls to the first entry starting with "N." The user can stop typing when the desired choice appears and select it from the list.

  If you attach a list to a field using its Field Properties in the Structure editor, the list will also appear when the field is selected in the Query editor. For more information, see the section "Creating New Fields" on page 172.

  You can also attach the list to the field as a data entry control in a form. The list will appear only when the field is selected in this input form, not in all forms or the Query editor.

  The figure below shows a choice list being displayed.



  If the number of elements that the list contains is limited, you may not need to use lists. For instance, in the case of a list that has only two values such as Male or Female, you may consider using a Boolean field. This would allow you to use interface elements such radio buttons or picture radio buttons. Even for cases that include four choices, you can use check boxes.

Hierarchical Lists

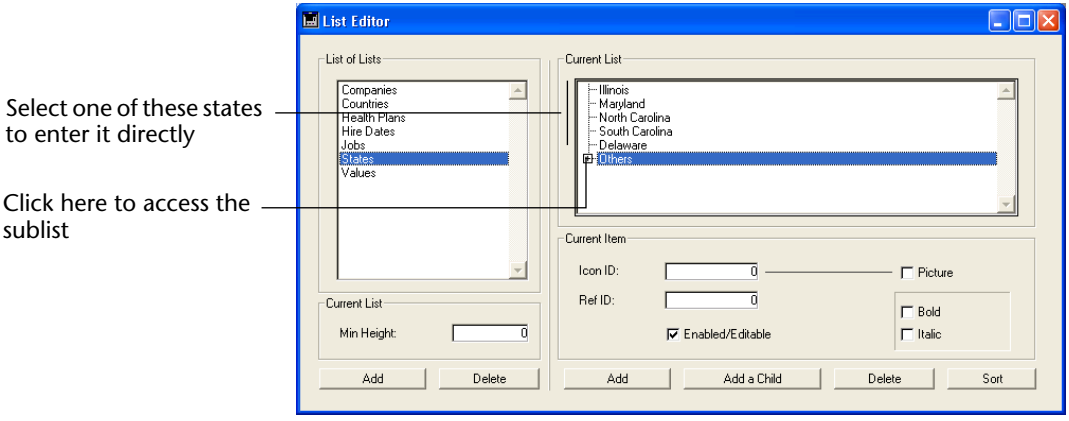
4th Dimension allows you to create hierarchical lists. Selecting an element from the parent list displays a sublist.

There is no standard way to use hierarchical lists; it depends on what you use them for. For example, a list of states would take many entries, thus slowing down the selection process. There are several approaches when you want to use hierarchical lists for this type of problems.

Besides their use for entry purposes, you can use hierarchical lists to populate the following objects:

- Hierarchical lists
- Tab controls
- Hierarchical pop-up menus
- Scrollable areas
- Combo boxes

You can often divide a list of values in two value categories. In the example mentioned above, you could determine which states are selected the most often. If 80% of the values selected refer to a handful of states, you can place those states in a list and place the remainder in a sublist, as shown below.



This allows the user to be able to directly select an entry in 80% of the cases, making the entry faster in most cases.

A different approach consists in organizing states in different regions. The first list that is displayed allows the user to select the region and the sublists allow him to select the state. In this case, each selection will require the user to select a region, followed by a state. It still is faster than selecting a state in a list of fifty states.

Required and Excluded Values

Some data entry tasks are not crucial. If you enter a value that does not appear in a list, it may be perfectly acceptable. However, you may have an application in which an entry must be one of the values in a list. Any different value would cause a serious consequence such as delaying bill payment.

4th Dimension allows you to make a list required as part of the data entry controls on a form. This type of data entry control prevents a user from entering any value other than the ones in the list. For example, your company may have a specific group of job titles that are allowable in a personnel database.

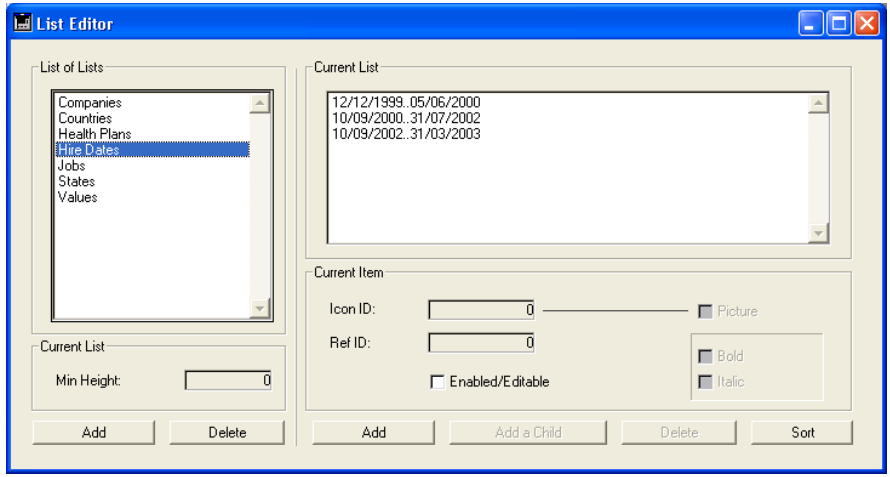
Another data entry control makes it possible to exclude the values in a list. The user then cannot type in a value that should not appear in the field. For example, your company may be prevented from doing business in certain countries. Placing them in an excluded list prevents them from being entered.

Non-Sequential Ranges of Values

One of the most useful data entry controls is the Maximum and Minimum setting for a number, date, or time field. Setting a maximum and minimum value prevents a user from entering a value outside this range.

Suppose you have three acceptable ranges for the field. You can use a list to create such non-sequential ranges. If you then make this list required for a field, values outside the three ranges are not accepted.

The figure below shows a list of ranges:



Conversely, you could create a list that specifies the ranges that are not valid. If you then make this list an Excluded list for a field, any entry within the ranges is not accepted.

When defining ranges, you should enter two dots ".." between the minimum and maximum values, otherwise 4th Dimension will not consider them as ranges but as a string value.

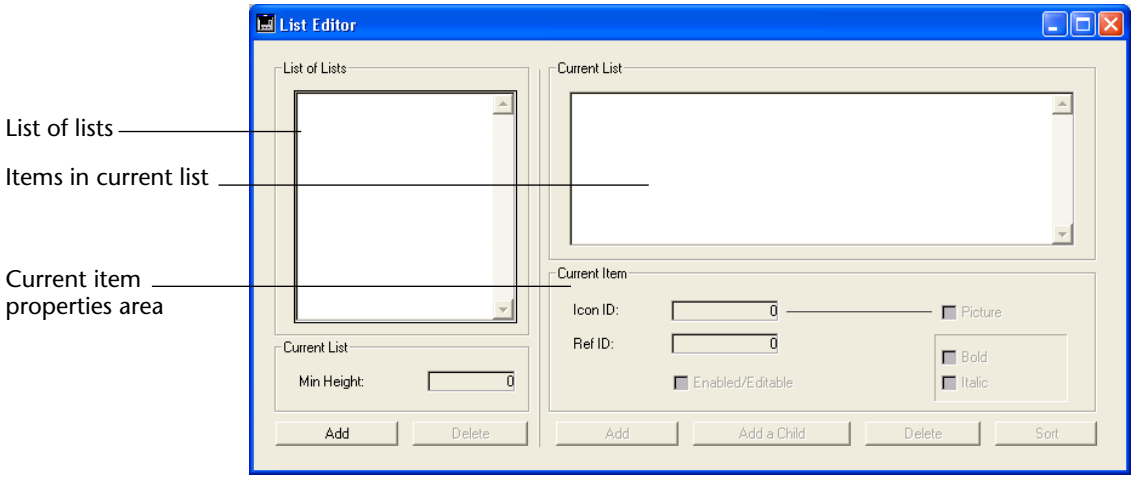
Creating Lists

You create lists with the List editor. You can modify any list at any time by returning to the List editor and making changes.

► To create a list:

1 **Choose List Editor from the Tools menu.**

The List editor appears.



The List editor displays the names of existing lists on the left. On the right side of the editor, a list of items in the selected list is displayed. The menu bar provides two menus: **Lists** and **Items**.

2 **Choose New from the Lists menu.**
OR
Click Add.
4th Dimension creates an empty frame in the Lists of Lists and displays an insertion point in the frame.

3 **Type the list name.**
You have created a new empty list. Now, you will create the items that will appear in the list as described next.

Adding Items to Lists

When you are adding items to a list, you can append new items to the end of the list or insert them anywhere in the existing list.

► To append items to a list:

1 **Select the name of the list to which you want to add items.**
If the list already contains items, they appear in the Current List area. If you want to insert a value in an existing list, select the value located after the value you want to insert. The new value will be created after that value.

2 **Choose New from the Items menu.**
OR
Click Add in the "Current List" area.

4th Dimension creates an empty frame in the Current List area and displays an insertion point in the frame.

3 **Type the item name.**

4 **To add additional items, repeat the above steps.**

5 **When you have finished, double-click the Control-menu box to save the current lists and close the editor window (click the Close box on Macintosh).**

Once the values entered, you can move the list items by drag and drop. You can also sort them by alphabetical order (see the paragraph "Sorting a List," page 652).

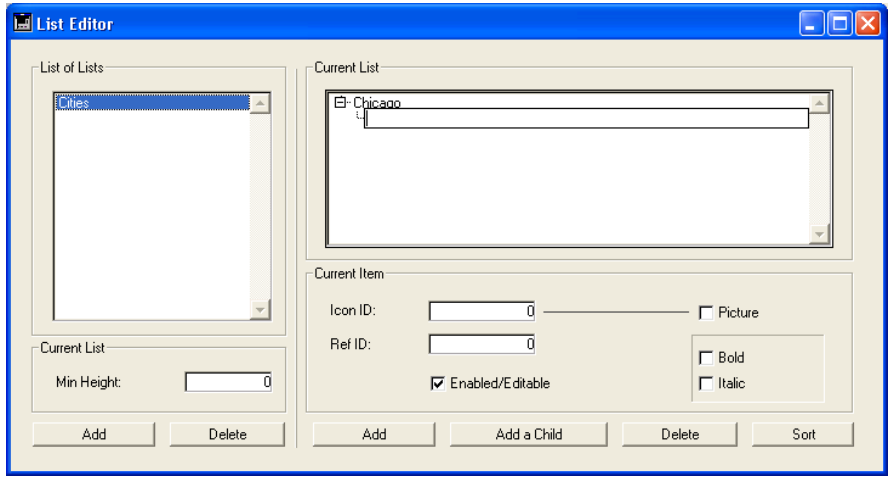
Creating a Hierarchical List

You can add a list to any list element. The hierarchy is not limited to two levels. You can use a hierarchical list to specify the contents of a hierarchical list interface object.

▶ To create a hierarchical list:

1 **Select the list item to which the sublist will be attached.**

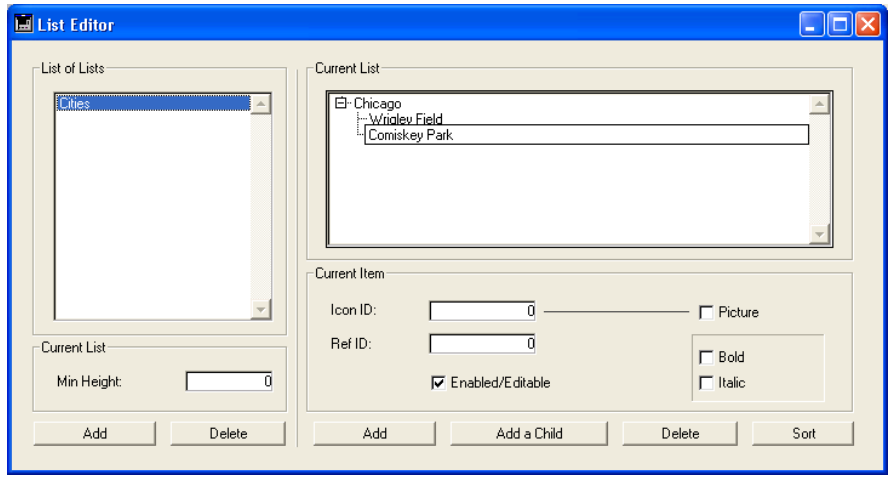
2 **Click Add a Child.**
OR
Choose New Child from the Items menu.
4th Dimension expands the selected list element and creates an entry area for the first item on the sublist.



3 **Enter the item.**

4 **To add another item to the sublist, click Add or choose New from the Items menu.**

Repeat these steps as necessary.



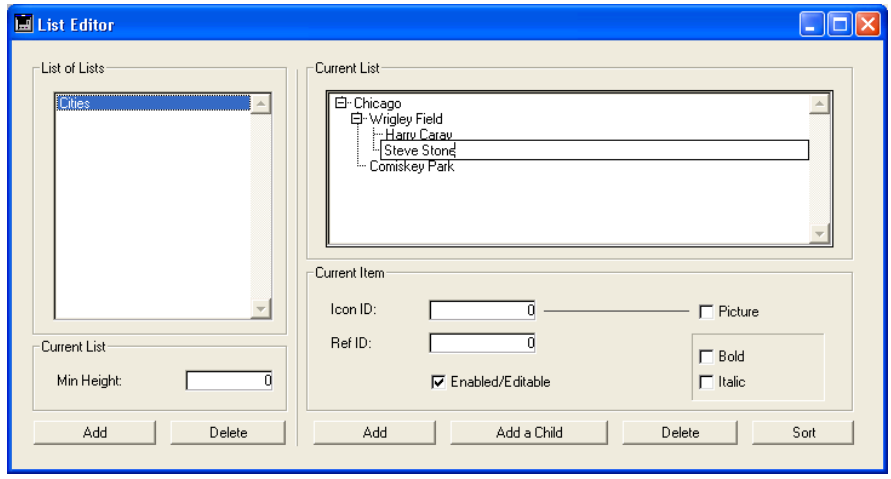
If desired, you can attach sublists to sublist items to continue the hierarchy.

► To attach a sublist to a sublist item:

1 **Select the sublist item.**

2 **Click Add a Child.**
OR
Choose New Child from the Items menu.

3 **Enter the item normally and repeat the process of entering items or attaching sublists to items, as desired.**
The following illustration shows a three-level hierarchy.



Deleting Items and Lists

You can delete items at any level of the hierarchy.

▶ To delete an item:

1 **With the List editor window open, select the list that contains the item you want to delete.**

2 **Select the item you want to delete from the Current List area.**
If necessary, expand the list.

3 **Click Delete in the Current List area.**
OR
Choose Delete from the Items menu.
4th Dimension deletes the item from the list and removes the space it occupied in the list.

▶ To delete a list:

1 **With the List editor window open, select the list you want to remove.**

2 **Click Delete in the List of Lists area.**
OR
Choose Delete from the Lists menu.
4th Dimension deletes the list and removes the space it occupied in the list.

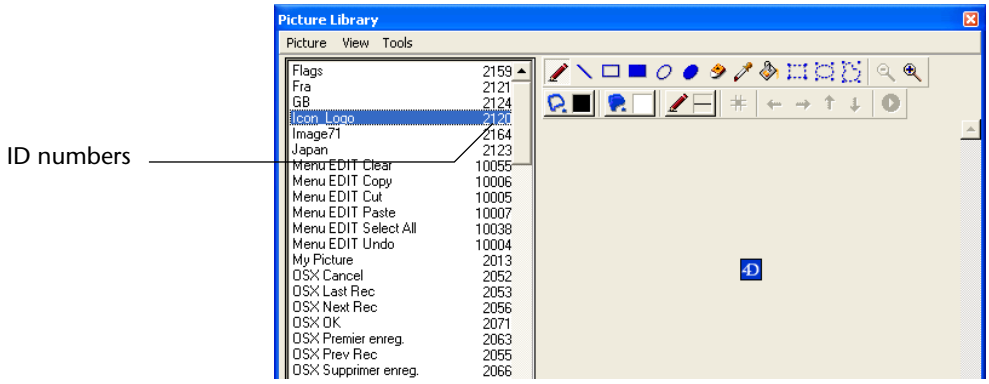
Adding a Small Icon to an Item

You can associate a small icon with an element in a list. When the list is displayed in a scrolling area or on a tab, this icon appears to the left of the value.

The small icons that you use are stored in the Picture Library.

Picture Library items

When you add a picture to the Picture Library, it automatically assigns it an ID number. You can use this number to associate the picture with an item in a list.



► To associate a small icon with an item:

1 **Select the item in the desired list.**

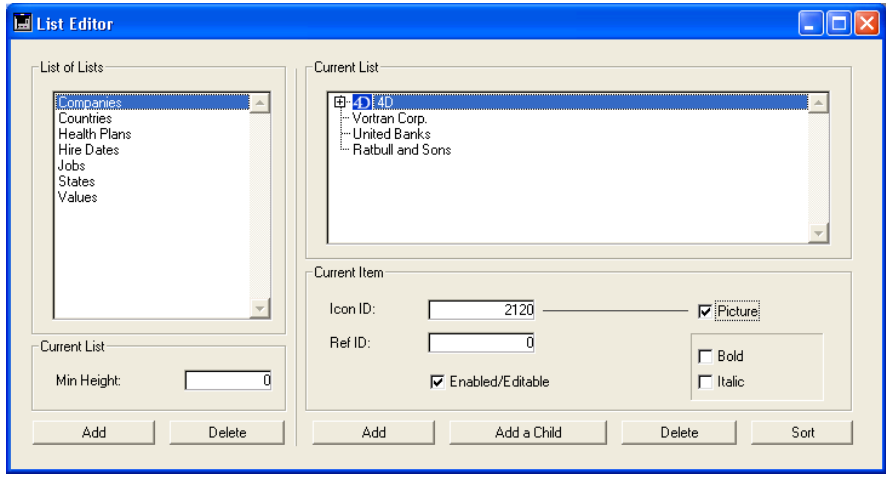
If necessary, first select the list from the list of lists and expand a list element. The "Current Item" properties area changes to show the properties of the selected item.

2 **Check the Picture check box and enter the ID number of the picture in the "Icon ID" area.**

3 **Press Tab to save the ID number.**

The Current List area changes to show the small icon associated with the selected list item.

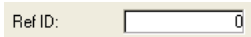
The following illustration shows the picture with the ID of 2120 (from the Picture Library shown above) associated with a list item.



Note Depending on the size of the icons you use, you may need to modify the list's height. For more information, refer to "Setting the Minimum Height of a List" on page 656.

Adding a Reference ID to an Item

The Current Item Properties area contains an entry area for the item's Reference ID ("**Ref ID**" area). The Reference ID is designed as a unique identifier for the item. It is of use only when you manage lists using methods.



When you need to use the language to determine which item in a list a user selects (e.g., which item in a hierarchical menu is selected), you can identify the user's choice using the Reference ID of the item. For more information, see the section "Hierarchical Lists" in the *4th Dimension Language Reference* manual.

Specifying Ranges in a List

4th Dimension allows you to enter ranges of numbers, dates, and times in a list. You can use these ranges as data entry validation ranges by making the list required or excluded in a form.

► To create ranges in a list:

1 **Create the list you want to use for ranges.**

2 **For each item, enter the minimum value of the range, two periods (..), and the maximum value.**

For example:

100..150

defines the range between 100 and 150, inclusive.

3 **Continue specifying ranges as separate items until you have defined as many as you need.**

Sorting a List

4th Dimension maintains the list of items in the order in which you enter them. You can sort the list alphabetically so that entries are more easily accessible to database users. Since a sorted list automatically scrolls to match characters typed at the keyboard, sorting usually makes data entry easier.

► To sort a list:

1 **In the List editor, select the list that contains the choices you want to sort.**

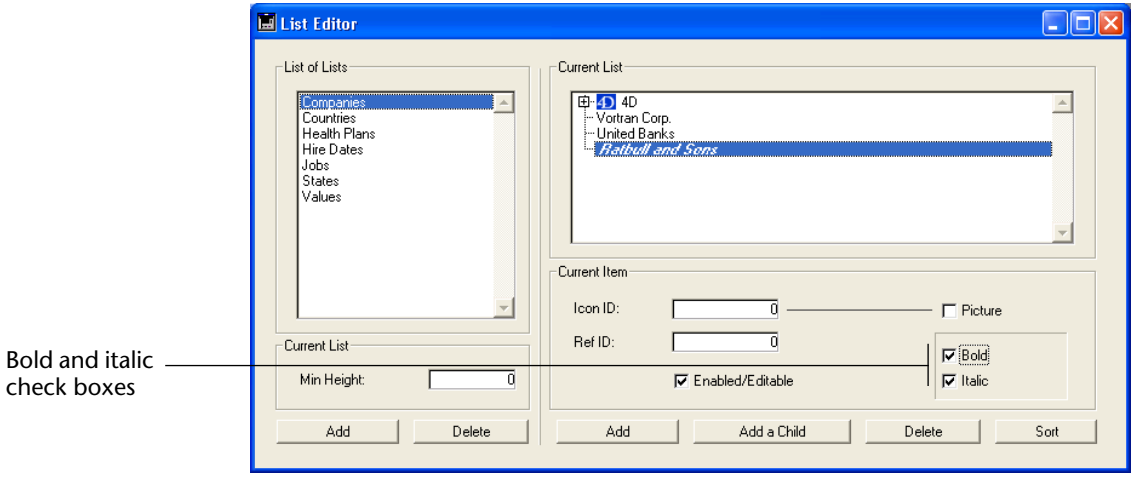
2 **Click Sort.**
OR
Choose Sort from the Items menu.

4th Dimension sorts the list in ascending order.

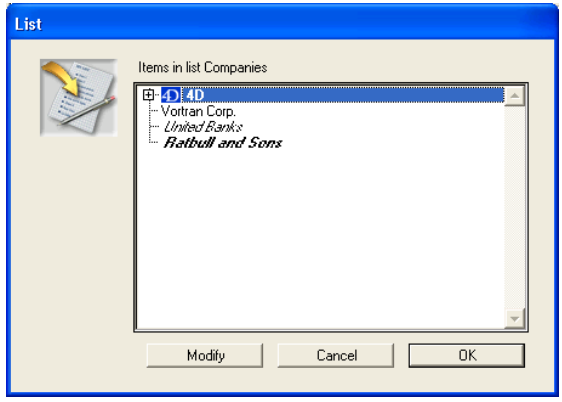
To sort the list in descending order, hold down the **Shift** key when you click the **Sort** button or when you choose the **Sort** menu command.

Specifying Font Attributes

When a list is used as a choice list, you can display list items in bold, italic, or bold-italic. The following illustration shows the bold and italic attributes applied to Chicago.



To apply font attributes, select the desired item in the Current List area and click the Bold or Italic check boxes. Check both check boxes if you want to use bold-italic. When the list is used as a choice list, the selected style attributes will be used, as shown in the following illustration.



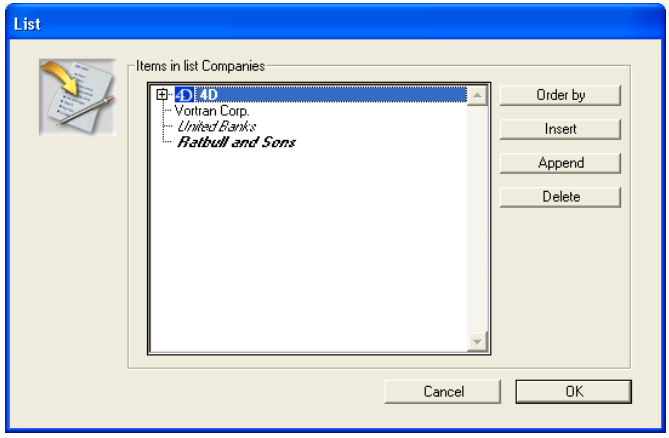
Making a Choice List Modifiable

4th Dimension allows you to specify whether a list of items can be changed by the user when the list is displayed as a choice list. By default, a list is modifiable. 4th Dimension places a check mark in the Lists menu to show that the list is user-modifiable.

If you allow a list to be user modifiable, the user has access to a special List editor in the User environment. The special List editor is for the assigned list only. The user cannot add lists, delete lists, or change any other list. If a list is modifiable, the user can make any change to that list's items.

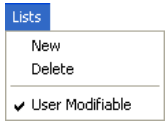
If a list is modifiable, in the User environment the **Modify** button is enabled in the List dialog box. When the user clicks this button, the List editor of the User environment is displayed.

The following shows the List editor in the User environment:



► To make the list user-modifiable:

1 Select the list that you want to make modifiable.



2 Pull down the Lists menu.

If **User Modifiable** has a check mark (√), the list is user modifiable. Release the mouse button without making a menu choice.

If **User Modifiable** does not have a check mark, go to the next step.

3 Choose User Modifiable from the Lists menu.

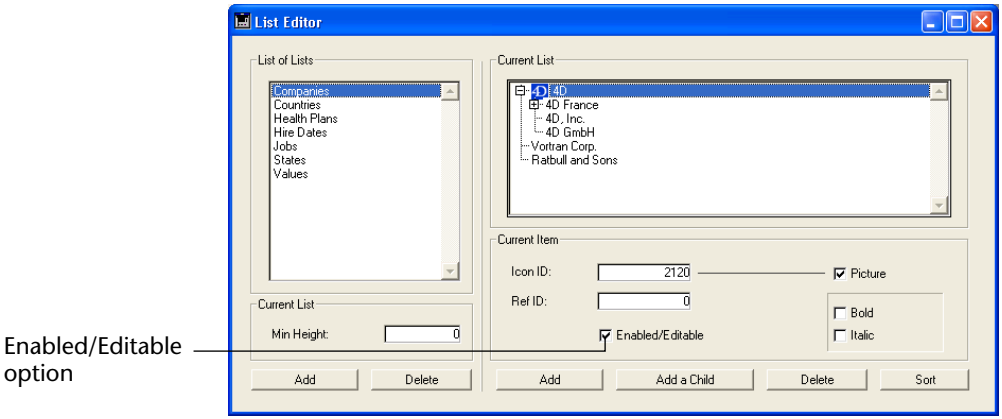
4th Dimension adds a check mark to the menu command. The list can now be modified in the User environment.

To prevent the user from modifying a list, select the list and choose **User Modifiable** to remove the check mark.

Enabled/Editable option

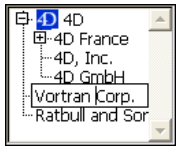
The List editor offers the **Enabled/Editable** option for each element of a list. It is checked by default.

This option is only used with lists displayed in the form of **hierarchical lists** or **tab controls**. Its action is different in both cases.



Hierarchical lists

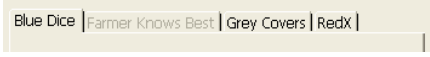
A list can also be used to specify the items in a hierarchical list. When the list is used in this manner, you can control whether each item in the list can be edited by the user. If a list item is editable, the user can hold down the **Ctrl** key under Windows (**Command** key under MacOS) and click on the item to get an insertion point. An editable item in a hierarchical list is shown below.



In this case, the **Enabled/Editable** option lets you allow the user to modify the element of the hierarchical list.

Tab controls

When a list is associated with tab controls, you can enable or disable each tab control that corresponds to an element of the list. A disabled tab control will be displayed in gray in the form. In the following example, the tab control “Farmer Knows Best” is disabled.



In this case, the **Enabled/Editable** option lets you enable the tab control corresponding to the element.

Setting the Minimum Height of a List

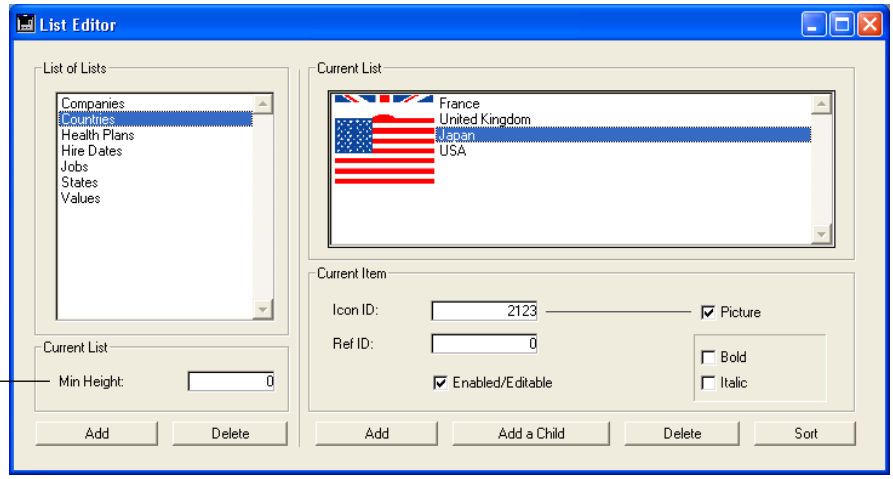
When 4th Dimension displays a list, it uses the font size of the hierarchical list object to determine the vertical spacing between adjacent list items. If you use a list to specify the values of a hierarchical list, you can specify a larger vertical spacing. The main reason you would want to do this is to provide additional space for icons that are attached to list items. Or, you can use this feature simply to spread out the list items.

► To specify a minimum height:

■ **Enter a value in points in the "Min. Height" entry area.**

The effects of this value are displayed immediately in the Current List area. The following illustrations show the effect of increasing the minimum height.

Minimum Height: 0 (default value)



Minimum height entry area

Minimum Height: 54

Note It may be necessary to resize the dialog box in order to view the effects.

Dragging a List into a Form

4th Dimension allows you to use shortcuts to create hierarchical lists or drop-down hierarchical menus that are associated with lists. These operations can be performed by dragging lists from the List editor.

- To insert a hierarchical list in a form, drag the list from the List editor to the form.
- To insert a hierarchical menu in a form, drag the list from the list editor to the form while pressing the **Shift** key.

Note You can use these shortcuts from the Explorer window.

12 Using the Picture Library

Use the Picture Library to store graphics that you can use as design elements in forms, as toolbar or list icons, picture menu items, or picture buttons. With the Picture Library, you can use a graphic in several places in your database but you need to store it in only one place. When you update an image in the Picture Library, all references to the image are updated automatically. This feature can reduce the size of your Structure files and make changes to the database easier to manage.

The Picture Library also includes a 2D Paint editor that allows you to create or retouch pictures. It is an ideal environment to design buttons and icons.

In addition to the features mentioned above, the Picture Library also includes integrated functions that can create or edit tables of thumbnails in order to create picture buttons or picture menus.

Managing and Viewing Pictures

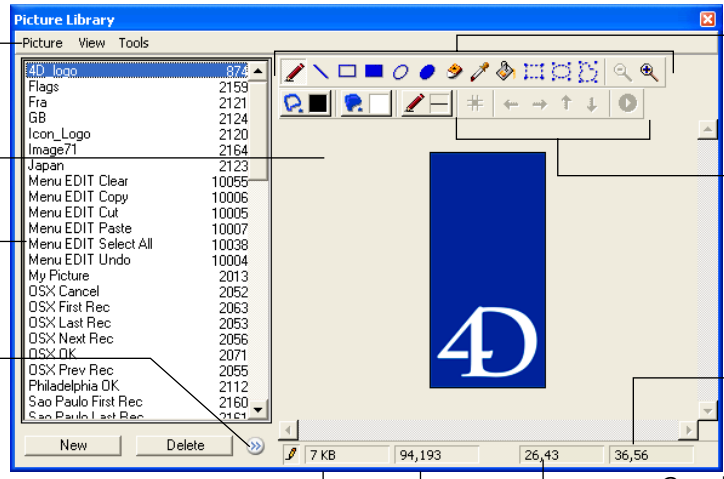
The Picture Library appears as a floating palette, which allows you to work simultaneously with different windows (for example, different forms) while keeping the library in the front.

► To display the Picture Library:

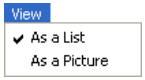
1 **Choose Picture Library from the Tools menu.**

The window contains menus that allow you to access picture management functions as well as a toolbar to edit the contents of the pictures.

You can also hide or show the picture's viewing/editing area by clicking on the switch at the bottom of the picture list.

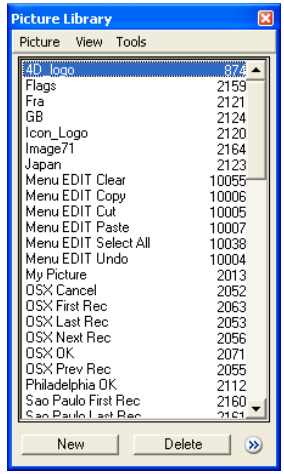


The different tools are described in the following sections.

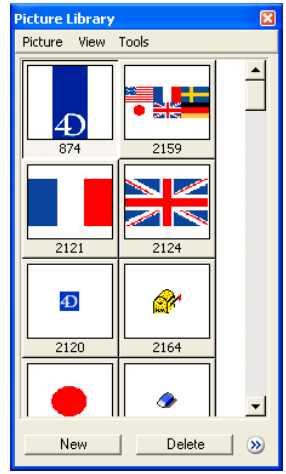


You can define how you want to view the images in the Picture List Area using the **View** menu:

View as a **List** (default mode)



View as a **Picture**



Note The List display mode must be selected if you want to use the drag and drop features of the Picture Library.

Adding Pictures

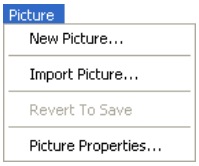
You can add pictures to the Picture Library in three ways:

- By importing a picture file
- By pasting a picture from the Clipboard
- By creating a new picture in which you draw its contents

Importing a Picture File

The Picture Library benefits from conversion routines provided by Apple® QuickTime® version 4 (or greater) that allow you to import a wide variety of picture formats under MacOS and Windows. In order to take advantage of this feature, you should install QuickTime 4 on your system.

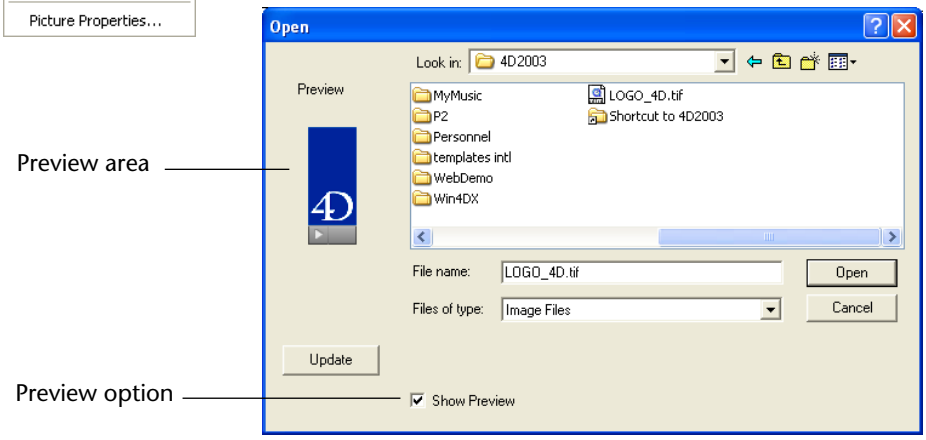
If QuickTime 4 is not installed on your system, only Mac type images (*.PCT or *.PIC) can be imported, under MacOS and Windows.



► To import a new picture into the Picture Library:

1 Choose Import Picture from the Picture menu of the Picture Library.

The following dialog box appears:



If QuickTime® version 4 (or greater) is installed on your system, this dialog box will allow you to open different picture types depending on the QuickTime converters available on the machine. You can also preview pictures by clicking on the **Show Preview** button:

If QuickTime 4 is not installed on your machine, only *.PCT or *.PIC files can be selected.

2 In the Open File dialog box, select the file and click Open.

The Picture Properties dialog is displayed. The photo's dimensions and name are automatically defined depending on the picture imported.

For more information on picture properties, refer to "Setting Picture Properties" on page 663.

Note Once the picture has been imported, it is converted to the internal library format. Future viewing of the picture from other machines will not require QuickTime.

3 If necessary, modify the ID number or other properties and click OK to create the picture in the Picture Library.

Note You can modify the ID number of the picture only when it is being created in the Picture Library.

Pasting a Picture from the Clipboard

► To paste a picture that is stored in the Clipboard:

1 Copy a picture to the Clipboard.

2 Select the Picture Library list area.

3 Press the Ctrl+v keys (Windows) or Command+v (MacOS).
OR
Choose Paste from the Edit menu.

The Picture Properties dialog is displayed. The dimensions that appear in that dialog are the picture's dimensions and its default name is *From ClipBoard*.

For more information on picture properties, refer to "Setting Picture Properties" on page 663.

4 If necessary, modify the ID number as well as other properties and click OK to create the picture in the Picture Library.

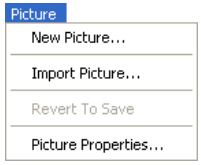
Note You can modify the ID number of the picture only when it is being created in the Picture Library.

Creating a New Picture

The Picture Library contains an integrated graphics editor. You can create a picture and draw its contents in that editor.

► To create a new picture:

1 **Click the Create button located under the Picture list area.**
OR
Choose New Picture... from the Picture menu in the Picture Library.



The Picture Properties dialog box is displayed. The dimensions that appear in that dialog are the default values for the picture's dimensions.

For more information on picture properties, refer to "Setting Picture Properties" on page 663.

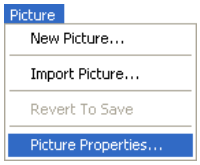
2 **If necessary, modify the ID number or other properties and click OK to create the picture in the Picture Library.**

Note You can modify the ID number of the picture only when it is being created in the Picture Library.

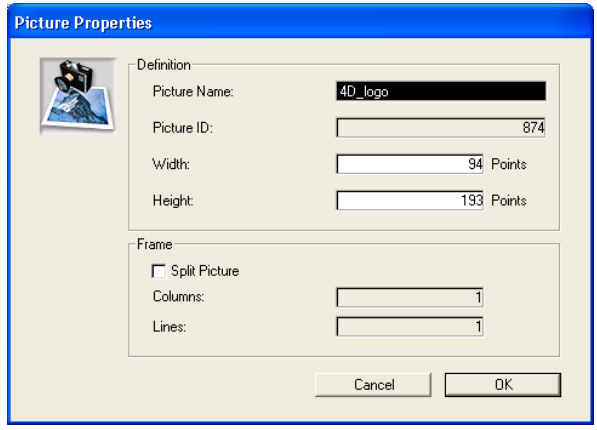
Setting Picture Properties

The Picture Properties dialog box allows you to set or display the the picture's name, ID number, dimensions, as well as its frame attributes.

No matter how you add a picture, the dialog box shown below appears.



You can display this dialog box at any time by selecting a picture and choosing the **Picture Properties...** command in the **Picture** menu of the library or by double-clicking on the picture itself.



This dialog box displays the following properties:

- **Picture Name**: allows you to modify the default name of the picture. You can assign the same name to several pictures, only the ID number has to be different.
- **Picture ID**: allows you to assign a unique ID number to the picture. This number is the reference number for the picture. It is how you refer to the picture when creating picture buttons, picture pop-up menus, custom toolbars or when you handle pictures programmatically.

Note You can define this number when you create the picture, but you cannot modify it afterwards.

- **Width** and **Height**: allow you to define the size of the picture. These values are precalculated when you import a picture (from a file or from the Clipboard). When you split the picture (see below), the values correspond to the size of each frame.
- **Frame area**: allows you to create thumbnails from a single image for use in creating an array of buttons or picture menus. For more information, please refer to the paragraph "Creating Thumbnails", page 666.

To create the picture, accept the dialog box. Once the picture's properties have been defined, you can modify them at any time by selecting the picture and by choosing **Picture Properties...** from the library's **Picture** menu.

Creating and Modifying Pictures

The Picture Library has a tool bar (like a Paint program) that allows you to draw and modify pictures.

To create or modify a picture, you just have to click on the view/edit area.

The Picture Library's editor works only in bitmap mode. Of course, you can import a picture — from a file or from the Clipboard — of a different type (for example, a vector graphic) and its characteristics will be retained when you use it in your database. On the other hand, if you modify this picture in the editor, it will be transformed into a bitmap. In this case, its original characteristics will be lost when the modified picture is saved.

When you modify an imported picture in the library, an alert dialog box appears to tell you that the picture will be converted into a bitmap (canceling at this point leaves the picture untouched).

The Picture Library's graphic tools are the following:

Icons	Tools	Function	Option keys
	Pen	Draws point by point	Alt (Option): allows you to take the color above which the cursor is located
	Line	Draws a line	Shift: the angles are in multiples of 45°
	Hollow Rectangle + Full Rectangle	Draws a hollow rectangle Draws a full rectangle	Shift: draws squares Ctrl (Command): the rectangle is drawn from its center
	Hollow Oval + Full Oval	Draws a hollow oval Draws a full oval	Shift: draws circles Ctrl (Command): the oval is drawn from its center
	Eraser	Erases by using the current background color	
	Color Picker	Modifies the line color by using a color from the picture	
	Flood Fill	Fills an enclosed area with the current background color	
	Selection tools	Create a selection	
	Zoom	Zoom the picture	
	Outline Color + Fill Color	Line color and background color menus	These options can be modified from the editor's contextual menu (Click with the right mouse button under Windows or Control+click under MacOS)
	Line width	Line width menu	

You can also use the standard editing commands (such as copy, paste, etc.) in the **Edit** menu, 4th Dimension's tool bar, or the standard keyboard shortcuts.

Note If you use the standard Paste command while the Picture editor area is selected, the Clipboard's contents is inserted into the currently edited picture. If you use the standard Paste command while the Picture list is selected, a new picture is added to the list.

Saving and Cancelling Modifications

Any modification made to a picture is automatically saved in the library as soon as you click outside of the Picture editor area (which means as soon as the editor area "loses" focus).

***WARNING:** Once the picture has been saved, you cannot revert any of the modifications that you have made to it.*

While modifying a picture, you can cancel the last modification made to it by choosing **Cancel** from the **Edit** menu (or the toolbar) in 4th Dimension, as well as the standard shortcut **Ctrl+z** (Windows) or **Command+z** (MacOS).

You can also cancel all the modifications made to a picture by choosing **Revert to saved** from the library's **Picture** menu.

Creating Thumbnails

The Picture Library contains a set of functions that allow you to create and modify a row by column array of pictures for use in picture buttons or picture menus. The array may be either one- or two-dimensional.

In the Picture Library, such an array is called "Frames." Elsewhere in 4th Dimension, an array of pictures may be called "thumbnails" or a "table" of pictures. The idea consists of splitting a picture using rows and columns; each cell is therefore considered a "frame" or "thumbnail." 4th Dimension takes care of displaying the correct frame in the picture button or picture menu according to the specified parameters (for more information refer to "Picture Buttons" on page 429 and "Picture Pop-up Menus" on page 441).

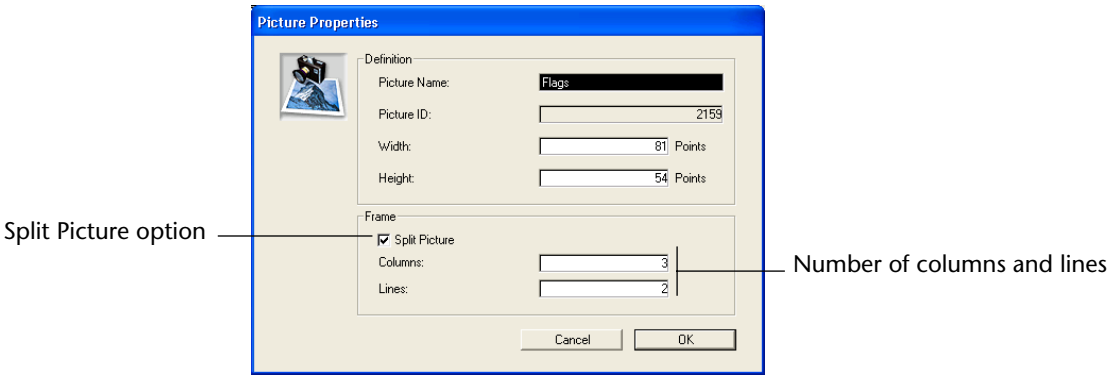
You can define a sequence of frames when creating a picture or even afterwards. You can also insert or delete lines, columns, or frames in an already defined sequence of frames.

Creating a Sequence of Frames

You can create a sequence of frames from a picture already placed in the Picture Library or when you add a picture to the Picture Library.

In both cases, you define how the picture is to be divided into separate frames in the Picture Properties dialog box. If you are creating a picture, the dialog box appears automatically. Otherwise, double-click on the picture or select it and choose **Picture Properties...** from the **Picture** menu.

The Frame area allows you to define the number of lines and columns of your frame sequence. To create frames, you must first check the **Split Picture** option:



Size of the Frames

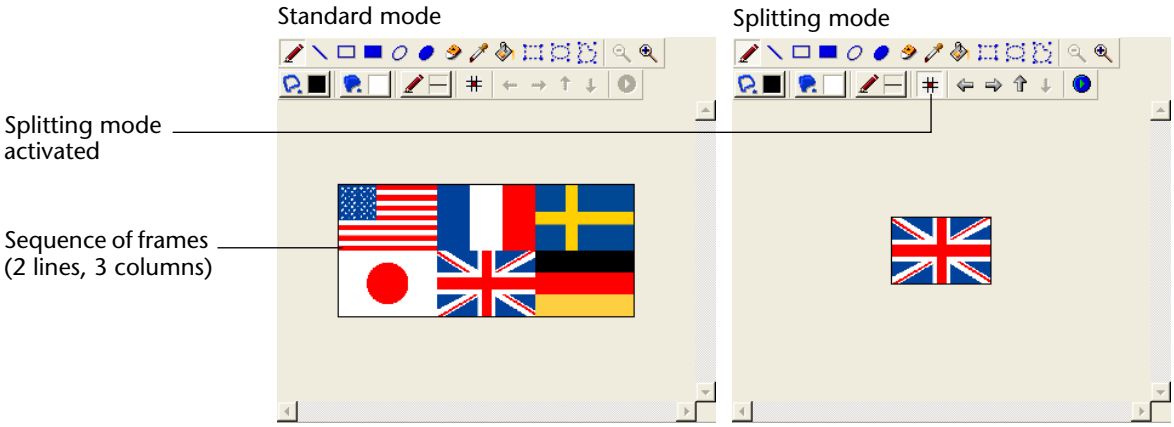
The size of the frames is automatically calculated by 4th Dimension. When you define a sequence of frames, the “Width” and “Height” areas are modified and the size of each frame is displayed.

If you want to modify the size of the frame later, you just have to enter new values into the Width and Height areas without worrying about the global size of the picture. Each resulting frame will be centered automatically (without the picture being distorted) in the new size if it is bigger. If the new size is smaller, each frame will be truncated.

Viewing the Frames

You can preview each frame to check the sequence's appearance by using specific tools in the Picture Library's tool bar:

When you activate the split mode, 4D splits the picture and displays the first frame (located in the upper left corner). You can then navigate through the frames using the arrow buttons.



You can also display all the frames automatically. This function is particularly useful if you want to create picture buttons that display in a continuous sequence. To do so, click on the **Test animation** button.

The frames appear in a continuous manner. To stop the test, click the **Test animation** button.

Inserting and Deleting Frames

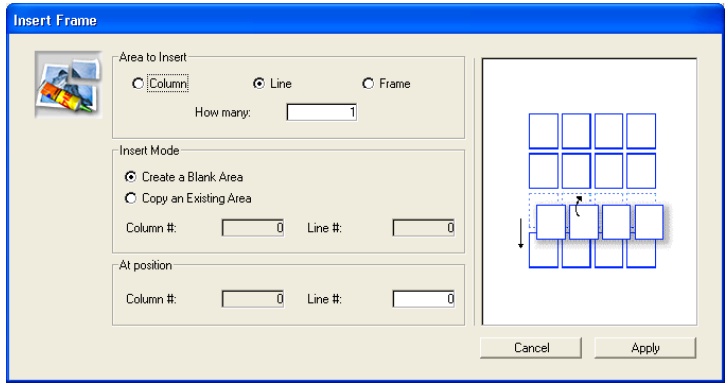
The Picture Library allows you to insert and delete frames in a previously created sequence of frames. You can insert frames as well as columns and lines.

► To insert frames:

1 **Select the frame sequence to modify.**

2 **Choose Insert... from the Picture Library's Tools menu.**

The following dialog box appears:



3 **Indicate whether you want to insert a column, line, or frame and indicate how many in the upper portion of the dialog box.**

The preview area on the right side of the dialog box shows you an example of the selected operation. Note that inserting an element moves the others (no element is replaced).

4 **In the "Insert Mode" area, indicate if you want the inserted element to be blank or if it should contain the contents of an existing element.**

In the latter case, you must designate the element to recopy.

Note The number of the first column and first line is 0.

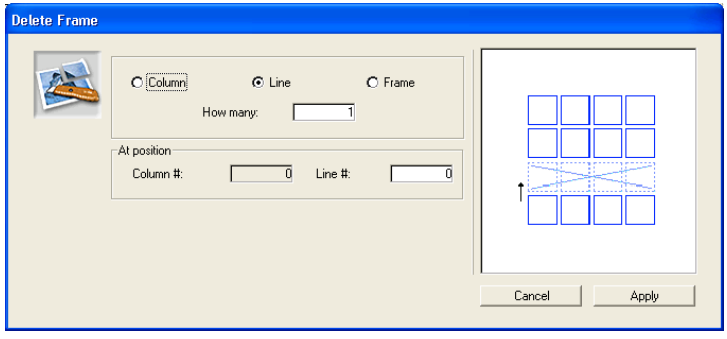
5 **Indicate the location in which you want the new element to be inserted and click Apply.**

The element is then inserted in the picture.

► To delete frames:

1 **Choose Delete... from the Picture Library's Tools menu.**

The following dialog box appears:



2 **Define the element (column, line, or frame) that you want to delete and indicate how many.**

The preview area on the right of the dialog box shows you an example of the selected operation.

3 **Indicate the position where you want the element to be deleted and accept the dialog box.**

Note The number of the first column and first line is 0.

Shortcuts for Inserting Picture Buttons and Pop-up Menus

When a picture is defined as a sequence of frames, you can use shortcuts to insert it as a picture button or a picture pop-up menu in your forms (for more information refer to "Picture Buttons" on page 429 and "Picture Pop-up Menus" on page 441). If you want to be able to use these shortcuts, you need to display the pictures as a list (and not as a picture).

- To create a picture button, drag the picture from the library and drop it in the form.
- To create a picture pop-up menu, drag the picture from the library and drop it in the form while holding down the **Shift** key.

Dragging and dropping a picture that is not defined as a sequence of frames will cause it to be inserted as a standard picture.

Note In the Form editor, the Property List allows you to distinguish between two types of pictures: **Picture Library** (dynamically updated when the source picture in the library is modified) and **Static Picture** (not associated to the library's source picture). Refer to the paragraph "Dissociating a form Picture from its Library Source," page 349.

13 Managing Processes

You can increase the functionality of a database by taking advantage of 4th Dimension's multi-tasking capabilities. In a multi-tasking system, database operations can run in separate *processes* — separate 4th Dimension tasks — that operate independently and concurrently.

Multiple processes are executed at the same time, allowing 4th Dimension to carry out several operations simultaneously. For example, one process might print a selection of records while another process allows a user to enter new records. In custom applications, multiple processes are used to manage a multi-window interface. Each window is managed by a different process. The process has its own current selection of records — even if two processes access data in the same table. The number of processes that can be executed at the same time is limited only by available memory.

This chapter explains how to do the following:

- Start a new process,
- View process information including the name, number, status, and total amount of time used by each process,
- Control process execution,
- Control whether a process is visible to users,
- Specify the frontmost process,
- Debug a process.

Processes

A process can be thought of as a 4th Dimension task that is carrying out some action — searching for some records, printing a selection, allowing a user to enter data, and so on. The exact operation that the process performs depends on the method with which it is connected.

As a separate 4th Dimension task, each process has the following elements for data management:

- A current selection for each table,
- A current record for each table
- Process variables,
- Locked records.

In addition, each process can have the following elements:

- Current input and output forms for each table,
- A menu bar,
- One or more windows,
- One active window (the frontmost window).

For a complete list of process elements and information about creating windows and menu bars, refer to the *4th Dimension Language Reference* manual.

You may notice that all of these elements are found in the 4th Dimension User environment. All processes have the same basic properties as the User environment and allow you to perform the same operations that you can in the User environment.

However, instead of having to perform the operations directly in the User environment, processes allow you to use methods to specify the actions you want 4th Dimension to take.

Anything that can be done with the 4th Dimension language — any user interface that can be created or operation that can be performed — can be done from a process.

Having multiple processes open at once gives you the ability to perform different actions or work with different aspects of a database simultaneously.

Opening multiple processes allows the user to do the following:

- **Work with more than one active window** You can have several active windows open at the same time. For example, you can enter data in one window and receive messages from colleagues in another window.
- **Work with more than one current selection at a time** Because each process has its own current selection, each process can have a different current selection from the same table. For example, an Employees database might contain a list of employees and their occupations. In a single-process database, you could display all engineers at once, or all accountants at once, but you could not make both selections the current selection for the same table. In a multi-process database, you can display the records of all engineers in one process and the records of all accountants in another process.
- **Work with more than one current record at a time** Each process can have a different current record. For example, you might want to compare one employee to another using an input form. In a single-process database, you can display only one employee's record. In a multi-process database, you can display each employee's record in a different process.
- **Start a lengthy operation in a separate process** You can perform a time-consuming operation such as printing a large selection of records in a separate process while you continue to work on your database.
- **Work with more than one input or output form at a time** You can view data in several different forms at once. For example, you could display a selection of records in a standard output form in one process and in a special report form in another process.

Processes Created and Managed by 4th Dimension

4th Dimension automatically creates and manages the following processes which control the operation of 4th Dimension:

- **User/Custom Menus** This process controls the User and Custom Menus environments.
- **Design** This process controls the Design environment. It is created when you enter the Design environment. This process does not exist in compiled mode or in a database opened with 4D Runtime.
- **Cache Manager** This process controls flushing or caching data to disk.

- **$Runtime Explorer** This process manages the Runtime Explorer window when it is displayed as a floating palette. It is created as soon as the Runtime Explorer is opened.
- **$4D Compiler** This process manages the compiler integrated into 4th Dimension. For more information about this point, refer to the chapter "Compiling a Database," page 689.

The User/Custom Menus and Cache Manager environments are created automatically when you open a database. The Design process is created automatically when you enter the Design environment. The User/Custom Menus and Cache Manager process always appear at the beginning of the list of processes.

In addition, 4th Dimension creates and manages the following processes:

- Indexing process,
- On Event Manager process,
- Web server process.

Unlike user-created processes, the processes created by 4th Dimension are always running and cannot be frozen or aborted. For more information about viewing processes in the Process List editor, see the section "Using The Process List" on page 679.

For more information about the processes created by 4th Dimension, refer to the *4th Dimension Language Reference* manual.

Time-Sliced Execution

Since in reality more than one process cannot execute at the same time, when you open multiple processes, 4th Dimension slices the total processing time so that execution is divided between all open processes. Execution alternates between processes so rapidly that the processes appear to be executing simultaneously. For instance, processing time is split between the Design process, the User/Custom Menus process, and the Cache Manager so that some milliseconds might be devoted to the User/Custom Menus process, the next to the Design process, the next back to the User/Custom Menus process, and so on.

Starting a New Process

4th Dimension allows you to start your own processes from the User or Custom Menus environments.

Each process that you start can perform a different task or present a different aspect of the data contained in your database.

The functionality of the process can be enhanced by a user interface created using any of the editors in the Design environment or using the 4th Dimension language. For instance, you can display an input form in a process to allow a user to enter records.

▶ Two start a new process:

1 **Create a method.**

The specific operation that each process performs depends on the commands and functions in the method. For more information about the 4th Dimension commands, refer to the *4th Dimension Language Reference* manual.

2 **Specify that 4th Dimension start a new process when the method executes.**

You can tell 4th Dimension to start a new process in the following three ways:

- Using the New process command in another method,
- Using the Menu Bar editor,
- Using the Execute Method dialog box.

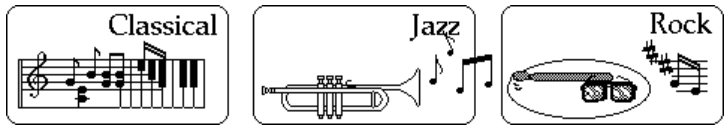
The alternative you select depends on what you are trying to do. Each alternative is described in detail in the following sections.

Starting a New Process Using New Process

There are many circumstances in which you might want to start a new process with the New process command. For instance, you might want to start a new process when a user clicks a button.You can do this by starting the process in the button's object method.

When you start a new process using the New process command, you should place the New process command in the method from which you want to start the new process. When that method executes, 4th Dimension starts a new process for the method specified in the parameters of the command and continues executing the original method.

The figure below shows a set of buttons in a compact disc database.



The object method for the **Jazz** button uses the New process command to start a new process for the method, *JazzSearch*:

myProcess:= **New process** ("JazzSearch";16000;"JazzRecords")

The new process, myProcess, carries out the actions specified in the method *JazzSearch*.

The method *JazzSearch* creates a selection of Jazz compact discs and displays them in a window. It contains the following statements:

```
QUERY ([Compact Discs];[Compact Discs]Category="Jazz")
If (Records in selection ([Compact Discs])>0)
   RefNo:=Open window (50;50;300;250;8)
   MODIFY SELECTION ([Compact Discs])
   CLOSE WINDOW
End if
```

When a user clicks the **Jazz** button, 4th Dimension opens a window and displays all the compact discs that contain Jazz music.

The window is running in the new process, myProcess.

If a different button is clicked, another process is started. If the user clicks the **Jazz** and **Rock** buttons, 4th Dimension starts two processes and displays the selection for each in its own process window.

The user can double-click records in either window to modify them in the current input form for that process.

For more information about the New process command, refer to the *4th Dimension Language Reference* manual.

Starting a New Process Using the Menu Bar Editor

You will often find that you want a new process to start whenever a user chooses a menu command. For instance, you might have a menu command that prints a group of records. Since printing records can be time-consuming, you specify that 4th Dimension start a new process whenever that menu command is selected.

► To start a new process using the Menu Bar editor:

1 Choose a menu bar in the Menu Bar editor.

2 **Select a menu from the Menus list.**

The menu commands and methods for that menu are displayed in the Menu Bar editor.

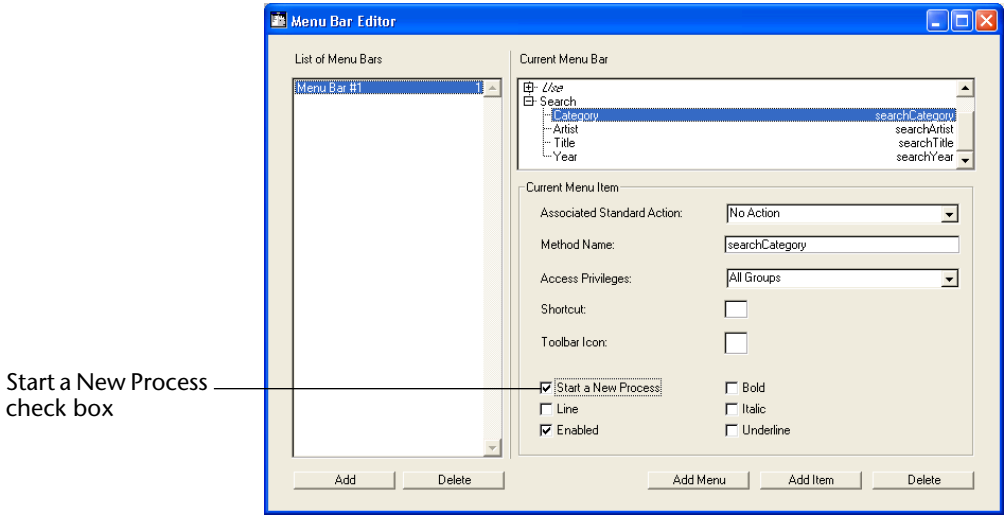
3 **Select the menu command for which you want to start a new process.**

When you select a menu command, the Current Menu Item area changes to show the properties of the selected menu item.

4 **Click the Start a New Process check box.**

This specifies that a new process should be started whenever that menu command is selected.

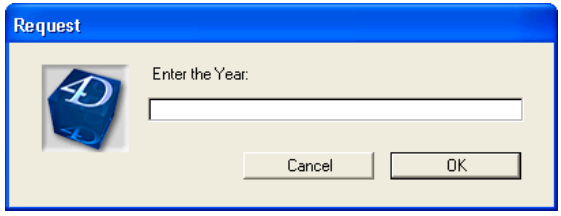
The following illustration shows a menu from the compact disc database:



The **Search** menu allows users to perform various searches. In the above example, the method for the **Year** menu command allows the user to enter the year. It then searches for all compact discs produced in that year and displays the selection in a window.

```
vYear:= Request ("Enter the Year:")
If (OK=1)
   QUERY ([Compact Discs];[Compact Discs]Year=vYear)
   If (Records in selection ([Compact Discs])>0)
      RefNo:=Open window (50;50;300;250)
      MODIFY SELECTION ([Compact Discs])
   End if
End if
```

When the user chooses **Year** from the **Search** menu in the Custom Menus environment, a dialog box appears requesting that the user enter a year.



When the user clicks the **OK** button, 4th Dimension displays the selection of compact discs produced the year the user entered. If the user chooses **Category** from the **Search** menu, the user can perform another search based on the type of music the compact disc contains.

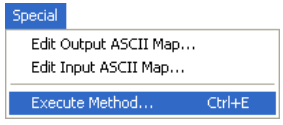
Starting a New Process Using Execute Method

You can choose to start a new process when you execute a method using the Execute Method dialog box in the User environment. One of the advantages to using this method to start a new process is that you can decide on a case-by-case basis whether you want to start a new process for a method.

4D Server You can choose to execute a process on 4D Server or on a 4D Client machine. Executing a process on the server lets you centralize the processes that do not require user intervention on the server machine. Executing a process on a specific client machine lets you allocate the processing. For more information, refer to the 4D Server *Reference manual*.

► To start a new process in the Execute Method dialog box:

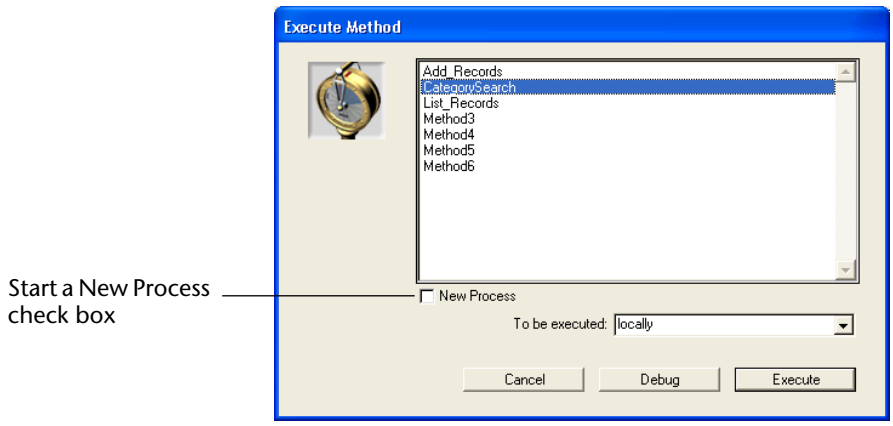
1 **Choose Execute Method from the Special menu in the User environment.**



The Execute Method dialog box appears.

2 **Select the method for which you want to start a new process.**

The figure below shows the *CategorySearch* method being selected:



3 **Click the "New Process" check box.**

Clicking the **New Process** check box tells 4th Dimension to start a new process when the method executes.

4 **Click the Execute button to execute the method.**

If you want to monitor or debug the execution of the method, click **Debug**.

4th Dimension executes the method within a new process.

Using The Process List

4th Dimension lists processes in the Process page of the Runtime Explorer editor window as soon as they are started. Each process is given a process ID, which is the same as the process number (discussed in the next section). You use this process ID to identify a specific process in commands and functions.

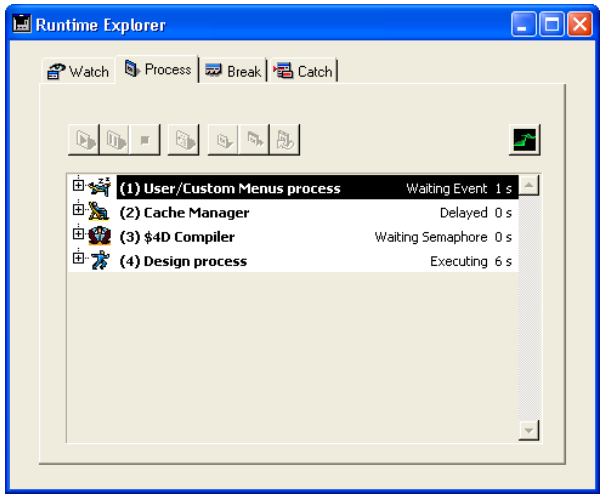
► To view the Process List:

1 **Display the Runtime Explorer Window.**

For more information, refer to "Runtime Explorer" on page 90.

2 **Click the Process Tab.**

Several processes are present by default (see the paragraph "Processes Created and Managed by 4th Dimension," page 673).



For each process, the Process List editor window gives the following information:

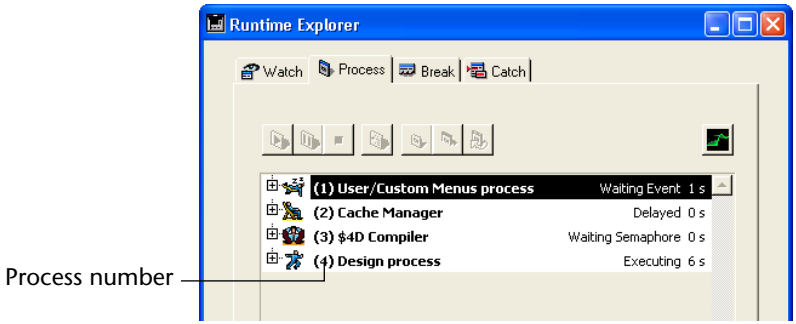
- Process number,
- Process name,
- Current status of the process,
- Total amount of execution time in seconds the process has taken since it was started.
- The graphical representation of the status and CPU time allocated to a process. For each process you can hide or display its graph by clicking on the icon located to the left of the process name.

The process attributes are explained in detail in the following sections.

4D Server 4D Client's Process page of the Runtime Explorer controls processes for a particular client. 4D Server's Process page of the Runtime Explorer controls processes for all clients connected to the server. For information, refer to the *4D Server Reference* manual.

Process Number

The default processes (User/Custom Menus Process, Cache Manager, $4D Compiler Process, Design Process and Web Server Process) are, respectively, numbers 1, 2, 3, 4 and 5[1].



When you start your own process, the process either appears as the next process in sequence or takes the place of a process that has been aborted.

For example, suppose processes 7 and 8 are executing. If process 7 is aborted, the next process to be started becomes process 7.

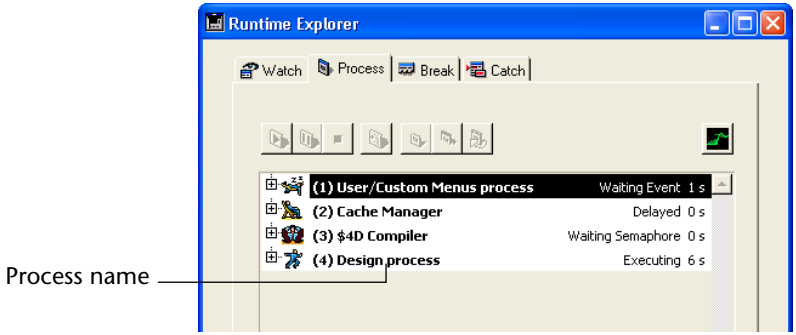
Note Processes are automatically aborted upon completion. You can abort a process before it has completed using the Process List editor. For more information about aborting a process, see the section "Aborting a Process" on page 685.

Process Name

If you start a new process using New process, you can specify its name as a parameter to the New process function. The name specified in the parameter appears as the process name in the Process List editor, as shown above.

1. If the Web server is launched at startup (see the Preferences), it then becomes number 4 and the Design Process is number 5.

For more information about the New process command, refer to the *4th Dimension Language Reference* manual.



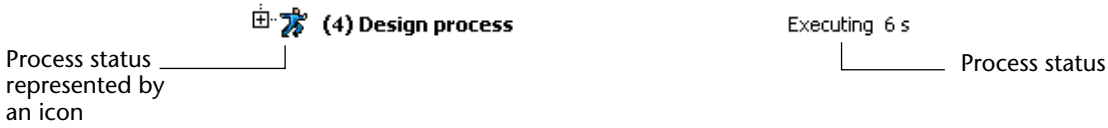
If you do not specify a process name using the New process command, 4th Dimension automatically assigns the process a default name. Default names are based on the method used to start the process, as follows.

- **Processes started from a menu command** If you start a process from a menu command, the process is given the default name "M_*ProcessNumber*." For instance, if process number 7 is started when a menu command is selected, the process is given the name "M_7".
- **Processes started from the Execute Method dialog box** If you start a process from the Execute Method dialog box, the method is given the default name "P_*ProcessNumber*." For instance, if process number 5 is started programmatically, the process is given the name "P_5".
- **Processes started using New process, but not explicitly named** If you start a process using a method but do not specify the name as a parameter to the New process command, the process name is left blank.

Note If the name of a process begins with a dollar sign ($), it is a local process that does not have access to tables or 4D Server.

Process Status

The status of a process is the current state of its execution—what the process is actually doing. In the Runtime Explorer the status of the process is indicated by the icon located to the left of the process name and by the text located next to the process name.



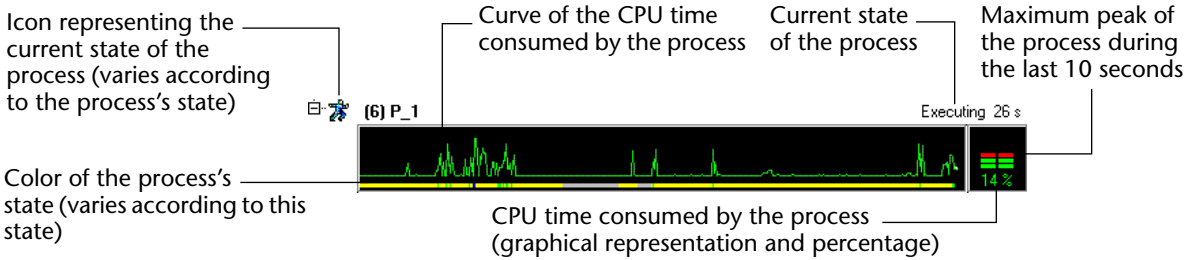
The following is an explanation of each status that can appear in the Process List editor window.

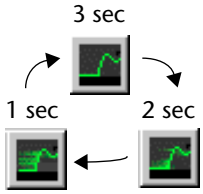
- **Executing** The process is currently executing.
- **Delayed** The process is delayed for a specific amount of time. During the period that the process is delayed, it does not take up any processing time. For information about how to delay a process, refer to the *4th Dimension Language Reference* manual.
- **Waiting event** The process is waiting for an action from the user such as a button being clicked or a menu command chosen.
- **Waiting for input/output** The process is waiting for some input or output to occur. For example, a process might need to wait while a group of records is being updated to disk.
- **Waiting for internal flag** The process is waiting for the cache manager to finish executing internal database tasks.
- **Paused** The process is paused until you tell it to resume execution. During the period that the process is paused, it does not take up any processing time. For more information, see the section "Pausing and Resuming a Process" on page 685.
- **Aborted** The process has been terminated. When a process is aborted, 4th Dimension frees any locked records, cancels any transactions opened by the process but not yet validated or canceled, and frees the current selection and current record. Processes are automatically aborted upon completion. You can also abort a process before it has completed by using the Process List editor. For more information about aborting a process, see the section "Aborting a Process" on page 685.
- **Hidden With Modal** A process which was displaying a modal dialog box has been hidden so that the user can no longer view the dialog box. The process will remain in this state until the dialog is shown.

Process Time

In managing processes, 4th Dimension divides processing time among existing processes so that no single process is executing at every moment. Thus, the process time is the total amount of execution time a process has taken (in seconds) since it started executing. Note that the process time does not reflect the total amount of time that has elapsed since the process started executing since, in reality, execution alternates between all open processes.

The Runtime Explorer displays the processing time for each process. If you expand the process graph, it displays the following information:





You can modify the frequency at which data should be updated — which can be every one, two, or three seconds. To do so, click successively on the icon in the top right portion of the window. The greater the update time is, the more CPU time is consumed by the Runtime Explorer process. The number of processes to graphically represent on screen also influences the CPU time consumed by the process.

Note No CPU time is consumed for a process when its graphical representation is closed.

When you click in the graphical area, a vertical line appears where you clicked and a tip indicates the state of the process at that instant. By holding down the mouse button and moving it from side to side, you can view the changes in the process's state.

The process management commands are now accessible by using the tool bar's buttons in the window: **Resume** , **Pause** , **Abort** , **Trace** , **Hide** , **Show** , **Bring to Front** .

Controlling Process Execution

The Runtime Explorer allows you to control the execution of processes by pausing, resuming, or aborting a process. These operations are covered in detail in the sections below.

Note You can also delay a process for a specific period of time. For more information about delaying a process, refer to the *4th Dimension Language Reference* manual.

Pausing and Resuming a Process

You can temporarily suspend the execution of a process by pausing it. You may want to pause a process to give other processes more execution time or to allow an event upon which the process depends to occur.

For instance, suppose you start a process that prints a selection of records. You then realize that you want to modify the data in one of the records so you first pause the process, finish your modifications, and then resume the process to continue printing the records.

► To pause a process:

1 **Select the process in the Process page of the Runtime Explorer.**

2 **Click the Pause icon**  **in the Process Page Toolbar.**

The status of the process in the Process List editor window automatically changes to "paused." The process remains paused indefinitely until you tell it to resume execution.

► To resume execution of a process:

1 **Select the process in the Process page of the Runtime Explorer.**

2 **Click the Resume icon**  **in the Process Page Toolbar.**

The status of the process returns to the status the process had at the time it was paused. For example, if the process was executing before it was paused, the process begins executing again. If the process was waiting for an event before it was paused, it continues waiting for an event.

Aborting a Process

A process is automatically aborted upon completion. However, you may need to abort a process before it completes for debugging purposes. Processes should not be aborted for any other reason. To stop the process from continuing execution, you abort the process in the Process List editor.

When a process is aborted, 4th Dimension frees any locked records, cancels any transactions opened by the process but not yet validated or canceled, and frees the current selection and current record.

► To abort a process:

1 **Select the process in the Process page of the Runtime Explorer.**

2 **Click the Abort icon**  **in the Process Page Toolbar.**

The status of the process in the Process List editor automatically changes to "aborted."

Tracing a Process

You can debug a process by monitoring its execution in the 4th Dimension debugger.

► To debug a process:

1 **Select the process you want to debug in the Process page of the Runtime Explorer.**

2 **Click the Trace icon**  **in the Process Page toolbar.**

If the process is being executed, the 4th Dimension Debug window appears, allowing you to debug the process by stepping through its execution and evaluating expressions such as the value of fields and variables used in the method.

If the Process was paused, 4th Dimension "stores" the request and displays the Debug window as soon as the execution of the process resumes. For information about using the Debug window, refer to the *4th Dimension Language Reference* manual.

You cannot debug the Cache Manager, Web Server, or Design processes.

Hiding a Process

You can make a process invisible in the User and Custom Menus environments by hiding it. When a process is hidden, any windows or menus created by the process are invisible to the user while the process is executing.

Hiding a process is useful for operations in which you open a window which you later want to close. Instead of aborting the process to close the window, you can make the window invisible to the user by hiding the process that opened it. Even though the window is hidden, the process continues to execute and complete the operation it began.

► To hide a process:

1 **Select the process you want to hide in the Process page of the Runtime Explorer.**

2 **Click the Hide icon**  **in the Process Page toolbar.**
The process is now hidden from view in the User and Custom Menus environments.

Note The process continues to execute even though it is hidden.

► To display again a process:

1 **Select the hidden process you want show in the Process page of the Runtime Explorer.**

2 **Click the Show icon**  **in the Process Page toolbar.**
The process is displayed again in the User and Custom Menus environments.

Bringing a Process to the Front

You can make a window the frontmost window by bringing its process to the front. For instance, if the User/Custom Menus Process is brought to the front, the User or Custom Menus environment is brought to the front of the screen.

You can bring any user-created processes to the front. If you have created a window for a process, the window becomes the frontmost window on the screen. If a menu bar is attached to the window, 4th Dimension brings the menu bar to the front of the screen and makes its menus the current menus. The current menu bar is replaced by the menu bar of the process that is brought to the front.

► To bring a process to the front:

1 **Select the process you want to bring to front in the Process page of the Runtime Explorer.**

2 **Click the Bring to Front icon**  **in the Process Page toolbar.**

Any windows attached to the process are brought to the front of the screen. In addition, 4th Dimension displays the menu bar for the front-most process window.

14 Compiling a Database

It is possible to compile your database, i.e., translate all of your methods into machine language. Compiling a database lets you check the consistency of the code and accelerate its execution, as well as making it possible to protect the code in its entirety. Compilation is an indispensable step between the development of databases using 4th Dimension and their deployment as stand-alone applications.

The compilation process is entirely automatic; however, compilation requires greater rigor when writing 4th Dimension code. The "Compiler Commands" section of the *4th Dimension Language Reference* manual provides advice and specific information concerning programming with a view to compilation. Furthermore, keep in mind that the compiler will indicate any programming errors and situate them in their context.

Introduction

This paragraph describes the theory of operation and the interests of database compilation.

What is a compiler?

The computer is a device in which commands are written using only "0"s and "1"s. This binary language is called *machine language*. The heart of the machine, the microprocessor, understands only this language. A program written in any high-level computer language (C, C++, Java, BASIC, 4th Dimension, and so on) is first translated into machine language, so as to be understandable to the computer's microprocessor.

There are two ways to do this:

- The statements can be translated during execution; the program is said to be *interpreted*.
- The statements are translated as a whole before program execution; the program is said to be *compiled*.

Interpreted mode

When a series of statements is executed using an interpreter, the process can be broken down as follows:

- The program reads a statement in the program's own language,
- It translates the statement into machine language,
- It executes the statement.

This cycle is executed for each of the statements in the program. The program that handles the execution of this kind of cycle is called the *interpreter*. For a database in the process of development, 4th Dimension methods are interpreted.

Compiled mode

A compiled program is translated in its entirety prior to execution. This process results in a new file that contains a set of statements in machine language. This set is saved for repeated use—the translation is performed only once and the compiled version of the program is available for repeated execution.

This phase is completely independent from any use of the program. The program that handles the translation is called the *compiler*.

Compiled objects

The compiler in 4th Dimension compiles the database methods, project methods, triggers, form methods and object methods in your database. If you do not have any of these elements in an application, the compiler will have nothing to compile.

When you have successfully completed compilation, the use of the compiled database is identical to that of the original one.

MacOS and Windows

The compiler in 4th Dimension can be used to generate code for both the MacOS and Windows platforms. You can choose to generate code for either one or both of these platforms.

When both types of code are requested (**Always compile for both platforms** option in the Preferences), the compiler generates "fat binary" code, i.e., both PowerPC code and Pentium code. When the database is used in compiled mode, the code suited for the processor of the user machine will be run.

If the compiled database must be used with 4D Server, the appropriate code will be loaded and run on each of the client machines: PowerPC on a Macintosh and Pentium on a PC.

Why compile your database?

The first benefit of compilation is, of course, speed of execution. There are two further benefits directly linked to compilation:

- Systematic code check,
- Database protection.

Speed of execution

The increased speed is due to two characteristics of compiled code: direct code translation, once and for all and direct access to variable and method addresses.

- **Direct and final code translation**
  The code of the methods written in 4th Dimension will be, using the compiler, translated once and for all. The time required in interpreted mode to translate all the statements is saved whenever you use a compiled database.

  Here is a simple case that illustrates this point. Take the case of a loop containing a sequence of statements that is repeated 50 times:

  ```
  For (i;1;50)
     `Sequence of statements
  End for
  ```

  In an interpreted database, each statement in the sequence is translated 50 times. Using the compiler eliminates the translation phase for each statement. For every statement in the sequence, we save 50 translations.

- **Direct access to variable and method addresses**
  In interpreted databases, variables are accessed through a name. Therefore, 4th Dimension must access the name in order to obtain the variable's value.

  In the compiled code, the compiler attaches an address to each variable, writes the variable's address directly in the code, and goes directly to that address whenever necessary.

Notes
- Operations requiring disk access may not be affected because their speed of execution is limited by the rate of transmission between the computer and its peripherals (drive or hard disk).
- Comments are not translated so they do not appear in the compiled code. Therefore, comments do not affect the execution time in compiled mode.

Checking your code

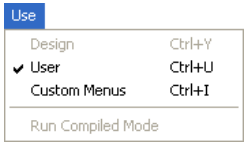
The compiler also operates as a syntax checker for your databases. It systematically checks your code and notes possible ambiguities, whereas 4th Dimension only does this when the method is executed.

Suppose that one of your methods contains a series of tests, as well as a sequence of statements to be executed. It is unlikely that you would fully test for all cases if the number of tests were very large. In this instance, a syntax error in an untested case might not show up until an end user encounters the case.

This sort of problem is avoided when you use a compiled database. When you compile a database, the compiler scans the entire database and analyzes each statement. The compiler detects any abnormality and generates an error message or warning.

Protecting your applications

Once you have compiled your database, you can use the application builder to erase the interpreted code. In this case, access to the Design environment is blocked. Therefore, the **Design** command in the **Use** menu is disabled for compiled databases.



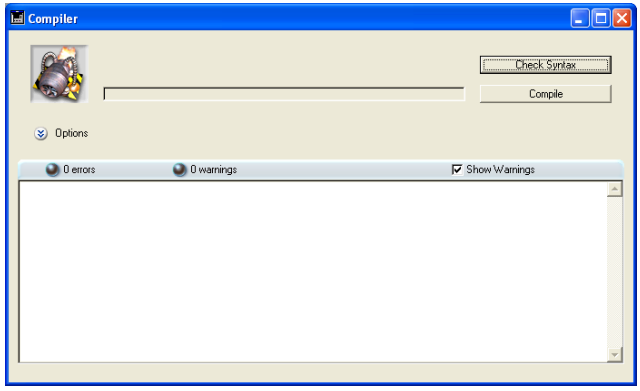
Note For more information about the application builder, refer to the chapter "Building Final Applications," page 711.

The benefits are:

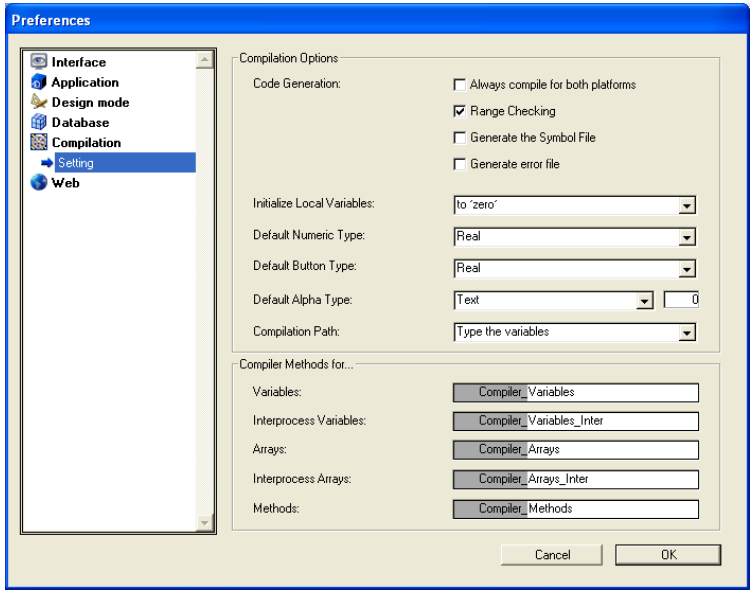
- The database structure cannot be modified, intentionally or by accident,
- Your methods are now protected.

Compilation in 4th Dimension

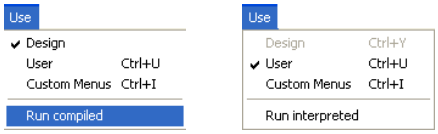
A compiler is integrated into 4th Dimension. Database compilation is carried out using the following dialog box:



Compilation is carried out in keeping with generic compilation options, configured in the Preferences dialog box:



Once the database is compiled, it is still possible to switch from interpreted mode to compiled mode, and vice versa, at any time via the **Use** menu, without having to quit the 4D application — except when the interpreted code has been erased (see the previous paragraph).

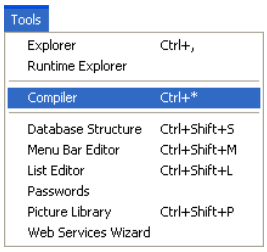


The database opening dialog box also allows the choice of interpreted or compiled mode on startup of the database (see the paragraph "Opening options," page 31).

4D Server It is not possible to compile a database from 4D Server or 4D Client (only the Check syntax function is available in 4D Client, see the paragraph "Compiler commands," page 695). Compilation can only be carried out using the 4th Dimension single-user version.

Compiler window

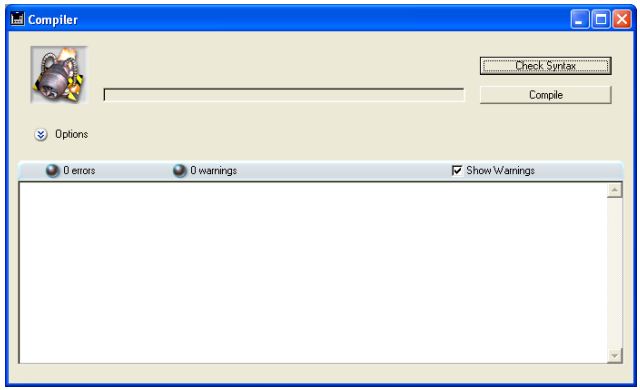
To display the compiler window, select the **Compiler** command in the **Tools** menu:



Note The **Compiler** command is disabled if the database does not contain at least one method.

This window is used for launching the compilation of the database (only with single-user 4th Dimension) or checking the syntax of the methods.

In addition, available options allow display or hiding of warnings, building/rebuilding of database typing methods and deletion of the compiled code.



Notes

- Database compilation requires an appropriate license. Without this license, it is not possible to carry out a compilation (the **Compile** button is inactive). Nevertheless, it is still possible to check the syntax and generate typing methods.
- With 4D Client, only the **Check syntax** and **Generate (typing methods)** buttons are active.

The commands of this window are described below.

Compiler commands

Check syntax

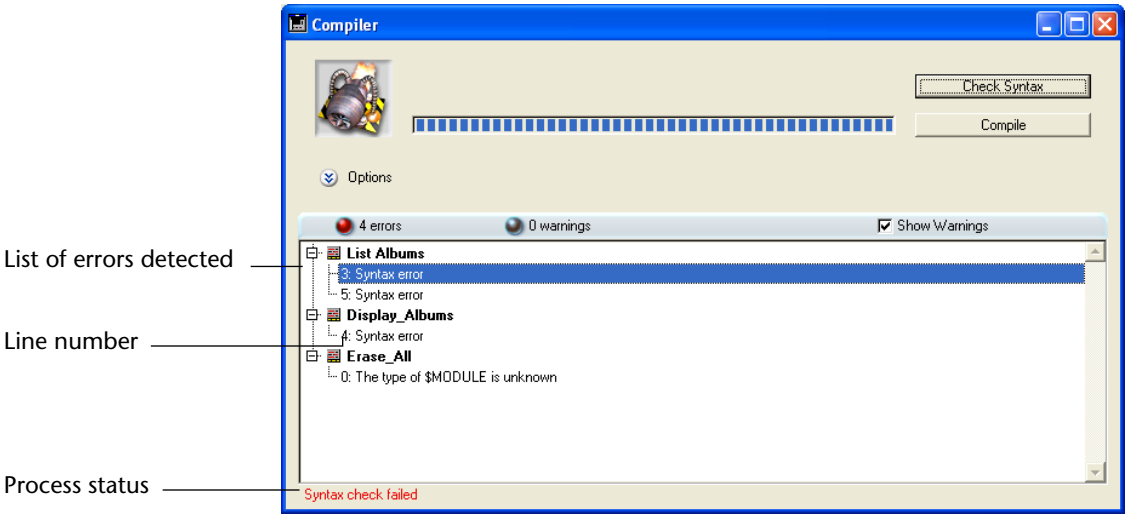
This button starts the execution of the syntax checking phase. At the end of checking, any errors detected are listed in the information area. As described in the following paragraph, it is possible to double–click on an error line in order to display the corresponding method.

This option is the only one available if you do not have a suitable license to allow the compilation of applications or when you use 4D Client.

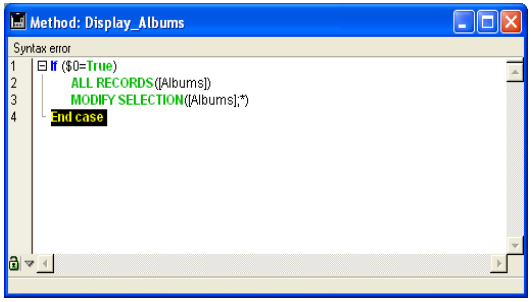
Compile

This button causes the immediate launching of the database compilation process. If the database has already been compiled, the new code compiled will replace the former.
Initially, different passes are carried out for checking, typing and initialization, in accordance with the configuration defined in the Preferences window (see the paragraph “Setting Page,” page 137).

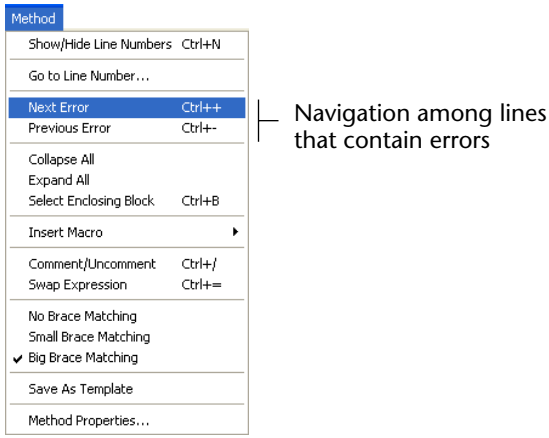
If no errors are detected, the actual compilation begins.
If errors are detected, the process is stopped and the information area of the window displays the method names and line numbers concerned in a hierarchical list:



Double-click on each error detected in order to open the method concerned directly in the 4D Method editor; the line containing the error is highlighted and the type of error is displayed in the syntax area of the window:



The **Previous error** / **Next error** commands of the **Method** menu of the editor allow you to navigate among the lines containing errors:



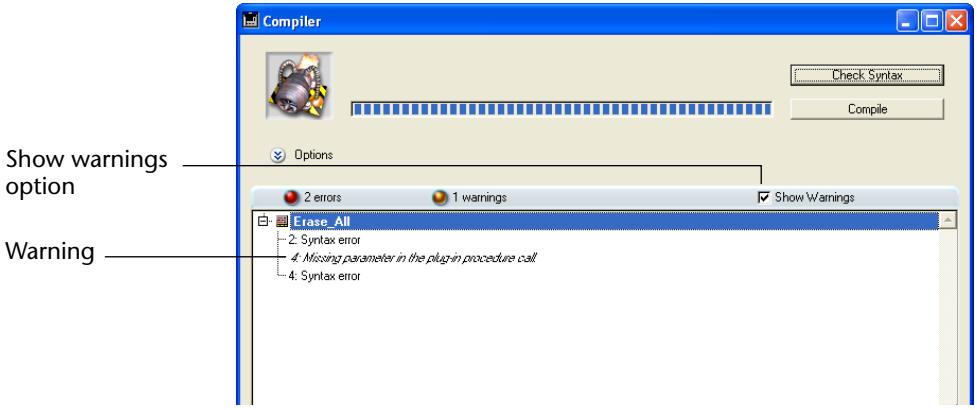
Note The number of errors found during your first compilations may be daunting, but do not let this put you off. You will soon discover that they often spring from the same source, i.e., non-compliance with certain database conventions. The compiler always provides a precise diagnosis of the errors in order to help you correct them.

Show warnings

Warnings are specific messages, generated by the compiler when it checks the syntax. These messages are intended to draw your attention to statements that might lead to execution errors. They do not prevent compilation. For more information about warnings, refer to the paragraph “Warnings,” page 705.

Depending on circumstances and the programming style used, these warnings may be more or less relevant. The **Show warnings** option can be used to display/hide them in the information area of the compiler window.

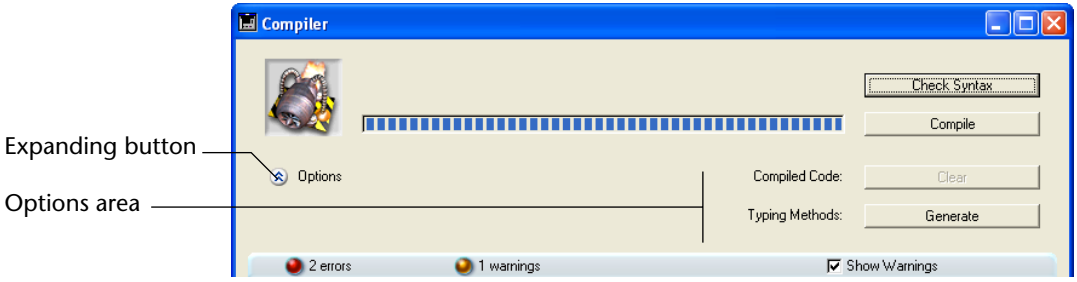
When this option is checked, the warnings (if any) are displayed in the window, after the other error types. They appear in *italics*:



Double-clicking a warning opens the corresponding method.

Options

The Compiler window offers two additional options (hidden by default). To display the options area, click on the expanding button:



This area contains the following buttons:

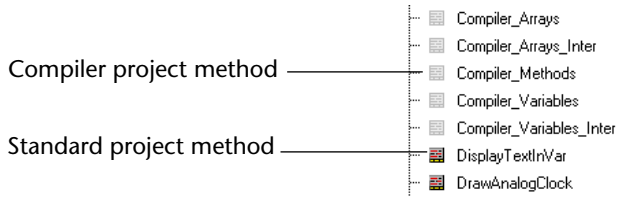
Clear (compiled code) This button is used to delete the compiled code of the structure file. When you click on it, all of the code generated during compilation is deleted. The size of the structure file will be reduced accordingly if you carry out compacting using 4D Tools.

The **Run compiled** command of the **Use** menu is then disabled and the "Open compiled" option of the database opening dialog box is grayed out.
Note that generated compiler methods are not deleted by this command.

Generate (typing methods)

This button creates (or updates) the typing “compiler methods.” Compiler methods are project methods that group together all the variable typing declarations, process and interprocess arrays, as well as the local variable declaration methods. These methods, when they exist, are used directly by the compiler during code compilation, which accelerates compilation. If these methods already exist, their contents are updated.

These methods, whose names must mandatorily begin with “Compiler_”, are generated by 4th Dimension. You can set the default name for each of the 5 compiler methods in the Preferences (see the paragraph “Compiler Methods,” page 140). In the Explorer, a specific icon designates the compiler methods that are generated and maintained by 4th Dimension :



Only the necessary compiler methods (i.e., those for which items already exist in the database) are built.

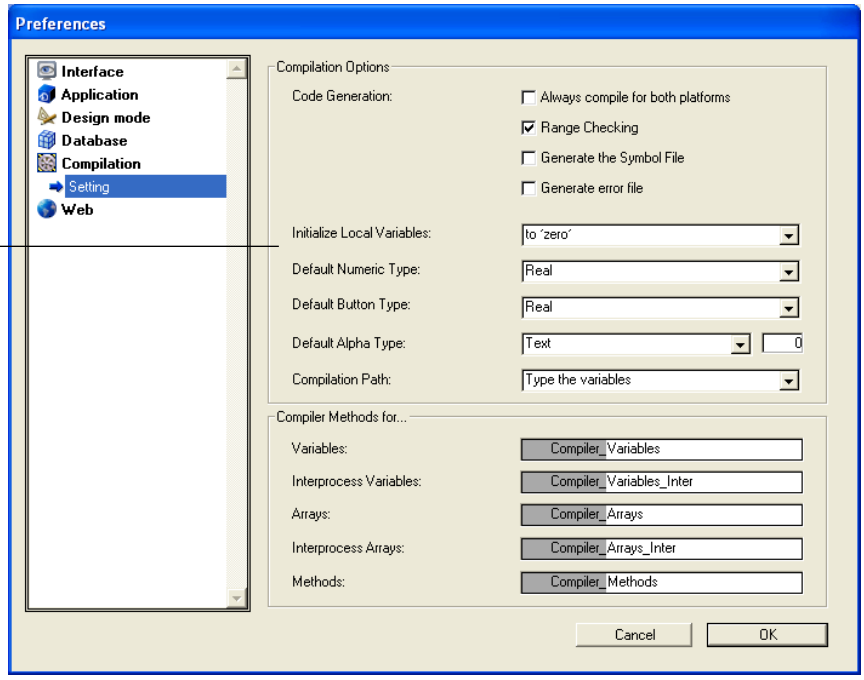
The information area indicates any errors found during method creation or updating. Double-clicking on an error line causes the method and line concerned to be displayed in the Method editor.

For more information about typing methods, refer to the “Compiler Commands” section of the *4th Dimension Language Reference* manual.

Compilation preferences

The generic compilation options are defined in the Preferences dialog box of the application. Options defined in this dialog box will be applied to all the databases opened using the current 4th Dimension application.

To access the compilation options, choose the **Preferences...** command in the **Edit** menu (Application menu under MacOS X). Then select the **Settings** page under the "Compilation" theme:



Generic compilation options

The compilation preferences are described in the paragraph "Setting Page," page 137.

Compilation diagnostic aids

There are three types of aids for the analysis and correction of databases:

- the actual analysis aid is provided by the symbol file. This table lets you find your way through your variables quickly. It is a valuable tool for interpreting the error messages reported by the compiler.

- the correction aid is provided by the error file which you can use as a text file.
- the execution aid or *range checking* provides you with an additional tool for monitoring the consistency and reliability of your applications.

Note Significant assistance is also provided for the typing of variables by the automatic compiler methods — see the paragraph "Generate (typing methods)," page 699.

Symbol file

The symbol file is a text type document whose length will depend on the size of your databases. By default, this file is not generated at the time of the compilation. To do so, you must check the corresponding option in the application (see the paragraph "Setting Page," page 137). When it is generated, the file is placed in the folder containing the database structure and is named:

- under Windows, DatabaseName.sym,
- under MacOS, DatabaseName.symb.

The symbol file is displayed as follows when it is opened using a text editor:

```
4D_Airports2003         10/9/2003       5:34 PM

<>4DCOMMENTS    Text            (M)  Compiler_Variables_Inter
<>4DQUERY       Text            (M)  Compiler_Variables_Inter
<>4DQUERYCOMMENTS       Text            (M)  Compiler_Variables_Inter
<>4DTABLETITLE  Text            (M)  Compiler_Variables_Inter
<>AAIRCRAFTDISTINCTRANGEKM      Long integer    1 dimension     (M)
Compiler_Arrays_Inter
<>AAIRCRAFTFAMILY       Text    1 dimension     (M)  PS_InitArrays
<>AAIRCRAFTID   Long integer    1 dimension     (M)  PS_InitArrays
<>AAIRCRAFTMANUFACTURER Text    1 dimension     (M)  PS_InitArrays
<>AAIRCRAFTNAME Text    1 dimension     (M)  PS_InitArrays
<>AAIRCRAFTRANGEKM      Long integer    1 dimension     (M)  PS_InitArrays
<>AAIRCRAFTSPEED        Long integer    1 dimension     (M)  PS_InitArrays
<>AAIRPORTGROUPID       Long integer    1 dimension     (M)  PS_InitArrays
<>AAIRPORTIATACODE      Fixed string:3  1 dimension     (M)  PS_InitArrays
<>AAIRPORTID    Long integer    1 dimension     (M)  PS_InitArrays
<>AAIRPORTLEVEL Long integer    1 dimension     (M)  GetCitiesofContry
<>ACAPITALAIRPORTS1     Text    1 dimension     (M)  Compiler_Arrays_Inter
<>ACAPITALAIRPORTS2     Text    1 dimension     (M)  Compiler_Arrays_Inter
<>ACAPITALAIRPORTSID1   Long integer    1 dimension     (M)
Compiler_Arrays_Inter
<>ACAPITALAIRPORTSID2   Long integer    1 dimension     (M)
Compiler_Arrays_Inter
<>ACITY Text    1 dimension     (M)  PS_InitArrays
<>ACOMPANYID    Long integer    1 dimension     (M)  PS_InitArrays
<>ACOMPANYNAME  Text    1 dimension     (M)  PS_InitArrays
<>ACOUNTRY      Text    1 dimension     (M)  PS_InitArrays
<>ACOUNTRYCITY  Text    1 dimension     (M)  GetCitiesofContry
<>ACOUNTRYCITYID        Long integer    1 dimension     (M)  GetCitiesofContry
```

The header displays the name of the database and the date and time of the document creation.

The document is divided into four parts:

- List of interprocess variables.
- List of process variables.
- List of local variables, in their method.
- Complete list of project methods and database methods with their parameters, if applicable.

List of process and interprocess variables

These two lists are divided into four columns:

- The first column contains the names of process and interprocess variables and arrays used in your database. These variables are listed in alphabetical order.
- The second column contains the type of the variable. Types are set by compiler directive commands or are determined by the compiler based on the use of the variable. If the type of a variable cannot be determined, the column is empty.
- The third column lists the number of dimensions if the variable is an array.
- The fourth column contains a reference to the context in which the compiler established the type of the variable. If the variable is used in several contexts, the context mentioned is the one used by the compiler to determine its type.
  - If the variable was found in a database method, the database method name is given as it has been defined in 4th Dimension, preceded by **(M)***.
  - If the variable was found in a project method, the method is identified as it has been defined in 4th Dimension, preceded by **(M)**.
  - If the variable was found in a trigger (table method), the table name is given, preceded by **(TM)**.
  - If the variable was found in a form method, the form name is given, preceded by the table name and **(FM)**.
  - If the variable was found in an object method, the object method's name is given, preceded by form name, table name, and by **(OM)**.
  - If the variable is an object in a form and does not appear in any project, form or object methods, nor any triggers, the name of the form in which it appears is given, preceded by **(F)**.

At the end of each list, you can find the sizes of the process and interprocess variables in bytes.

Note When compiling, the compiler cannot determine in which process a given process variable is used. A process variable can have a different value in each process. Consequently, all process variables are systematically duplicated as each new process is launched: it is thus advisable to watch out for the amount of memory that they will take up. Also, keep in mind that the space for process variables is not related to the stack size for the process.

List of local variables

The list of local variables is sorted by database method, project method, trigger (table method), form method, and object method, in the same order as in 4th Dimension.

This list is divided into three columns:

- The first column contains the list of local variables used in the method;
- The second column contains the type of the variable;
- The third column lists the number of dimensions if the variable is an array.

Complete list of methods

A complete list of your database and project methods is given at the end of the file, with the data types of their parameters and the returned result.

This information is presented in the following format:
Method name(parameter data types):result data type.

Error file

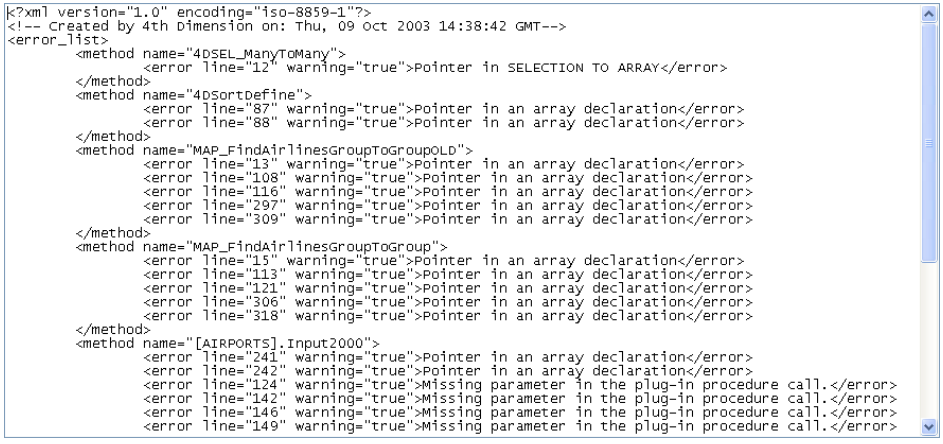
You can choose whether or not to generate an error file during compilation using an option located in the Preferences of the application (see the paragraph "Setting Page," page 137). When it is generated, the error file is automatically named DatabaseName.xml and is created:

- with 4th Dimension, next to the structure file of the database,
- with 4D Client, next to the .exe file of the 4D Client application (Windows) or next to the 4D Client software package (MacOS).

Although the errors are directly accessible via the compiler window, it can be useful to have an error file that can be transmitted from one machine to another, particularly within the framework of several different developers working together in a client/server environment.

The error file is generated in XML format in order to facilitate automatic parsing of its contents. It also allows the creation of customized error display interfaces.

The length of the error file depends on the number of errors and warnings issued by the compiler. When you open an error file using a text editor, it looks like this:



```
<?xml version="1.0" encoding="iso-8859-1"?>
<!-- Created by 4th Dimension on: Thu, 09 Oct 2003 14:38:42 GMT-->
<error_list>
        <method name="4DSEL_ManyToMany">
                <error line="12" warning="true">Pointer in SELECTION TO ARRAY</error>
        </method>
        <method name="4DSortDefine">
                <error line="87" warning="true">Pointer in an array declaration</error>
                <error line="88" warning="true">Pointer in an array declaration</error>
        </method>
        <method name="MAP_FindAirlinesGroupToGroupOLD">
                <error line="13" warning="true">Pointer in an array declaration</error>
                <error line="108" warning="true">Pointer in an array declaration</error>
                <error line="116" warning="true">Pointer in an array declaration</error>
                <error line="297" warning="true">Pointer in an array declaration</error>
                <error line="309" warning="true">Pointer in an array declaration</error>
        </method>
        <method name="MAP_FindAirlinesGroupToGroup">
                <error line="15" warning="true">Pointer in an array declaration</error>
                <error line="113" warning="true">Pointer in an array declaration</error>
                <error line="121" warning="true">Pointer in an array declaration</error>
                <error line="306" warning="true">Pointer in an array declaration</error>
                <error line="318" warning="true">Pointer in an array declaration</error>
        </method>
        <method name="[AIRPORTS].Input2000">
                <error line="241" warning="true">Pointer in an array declaration</error>
                <error line="242" warning="true">Pointer in an array declaration</error>
                <error line="124" warning="true">Missing parameter in the plug-in procedure call.</error>
                <error line="142" warning="true">Missing parameter in the plug-in procedure call.</error>
                <error line="146" warning="true">Missing parameter in the plug-in procedure call.</error>
                <error line="149" warning="true">Missing parameter in the plug-in procedure call.</error>
```

The structure of the error file is as follows:

- at the top of the file is the list of errors and warnings, sorted by method and in the order of their creation in 4th Dimension;
- in the ***General errors*** section, all the typing impossibilities and identity ambiguities are grouped together.

  These errors and warnings are listed using the following format:
  - first, the line number in the method (0 indicates general errors);
  - second, the warning attribute indicates whether the detected anomaly is a warning (warning="true") or an error (warning="false");
  - and third, a diagnostic that describes the error.

  If your database does not have any general errors, the file will not have a “General errors” section.

  An error file may contain three types of messages:

- Errors linked to a specific line,
- General errors,
- Warnings.

Errors linked to a specific line

These errors are displayed in context—the line in which they were found—with an explanation. The compiler reports this type of error when it encounters an expression in which it sees an inconsistency related to data type or syntax.

In the compiler window, double–click on each error detected in order to open the method concerned directly in the 4D Method editor with the line containing the error highlighted.

The list of syntax/typing diagnostic errors is found in the *4th Dimension Language Reference* manual.

General errors

These are errors that make it impossible to compile the database. There are two cases in which the compiler reports a general error:

- The data type of a process variable could not be determined.
- Two different kinds of objects have the same name.

General errors are so named because they cannot be linked to any specific method. In the first case, the compiler could not perform a specified typing anywhere in the database. In the second, it was unable to decide whether to associate a given name with one object rather than with another.

The list of general errors is found in the *4th* Dimension *Language Reference* manual.

Warnings

Warnings are not errors. Warnings do not prevent the database from being compiled; they simply point out potential code errors.

In the compiler window, warnings appear in *italics*. Double–click on each warning to open the method concerned directly in the 4D Method editor with the line concerned by the warning highlighted.

The list of warnings is found in the *4th Dimension Language Reference* manual.

Range checking

Range checking is checked by default in the Preferences of the application (see the paragraph “Setting Page,” page 137).

While all the other options operate during the compilation process, range checking begins when you run the compiled database. That is, range checking messages appear only while your compiled database is running.

Range checking furnishes additional analysis with respect to the quest for logical and syntactical consistency which normally characterizes a compiler. During range checking, the compiler poses the following question: "considering what you have requested, will the result that I am likely to obtain surprise you?" Range checking is an "in situ" controller; it evaluates the status of objects in the database at a given time.

Here is how range checking works. Suppose that you declared the array MyArray as Text. The number of elements in MyArray may vary depending on the current method. If you want to assign the value "Hello" to element 5 of MyArray, you would write: MyArray{5}:="Hello". If at that time MyArray has five or more elements, everything is fine. Assignment proceeds normally. However, if MyArray has less than five elements at that time, your assignment no longer makes sense.

A situation like this cannot be detected at compilation because of the presupposition that the methods are executing. The compiler would not know the circumstance in which this method is called. Only range checking enables you to monitor what is actually happening while your database is in use. In the above example, the compiler would display an execution error from within 4th Dimension. It is easy to see why range checking is especially valuable when arrays, pointers, and strings of characters are being processed.

The messages sent by the compiler when you request range checking are listed in the *4th Dimension Language Reference* manual.

Disabling range checking locally

Even when range checking has been enabled, there may be some cases where you prefer that it not be applied to certain parts of code that are considered to be reliable. More particularly, in the case of loops that are repeated a great number of times, and when running the compiled database on older machines, range checking can significantly slow down processing. Insofar as you have the *certitude* that the code concerned is reliable and cannot cause system errors, you can disable range checking locally.

To do this, you must surround the code to be excluded from range checking with the special comments `%R- and `%R+.

The `%R- comment disables range checking and `%R+ enables it again:

```
... `Range checking is enabled
   `%R-
... ` Place the code to be excluded from range checking here
   `%R+
... `Range checking is enabled again for the rest of the method
```

Note This mechanism will only operate when range checking is enabled.

Diagnosing Anomalies

Suppose you notice anomalies when running your databases. Before you start speculating about possible sources of these problems, remember the assistance provided by the compiler.

Potential anomalies are:

- 4th Dimension displays its own error messages.
  If possible, correct errors in your database according to instructions provided by 4th Dimension. If these are too general, compile your database again, making sure that the Range Checking option is enabled. Retest your database. At the location where the 4th Dimension message was displayed, you will see a more informative message from the compiler.
- Your compiled database does not perform exactly like your interpreted database. Take a closer look at the warning messages.
- Your database works in interpreted mode, but a system crash occurs in compiled mode. Make sure that you compiled the database using the Range Checking option and check to see if your compiled database is using the same plug-ins as the ones you used when compiling.
- Number or String variables do not return expected values. Check the default typing options in the Preferences and examine the symbol file to check that all your variables are typed properly.

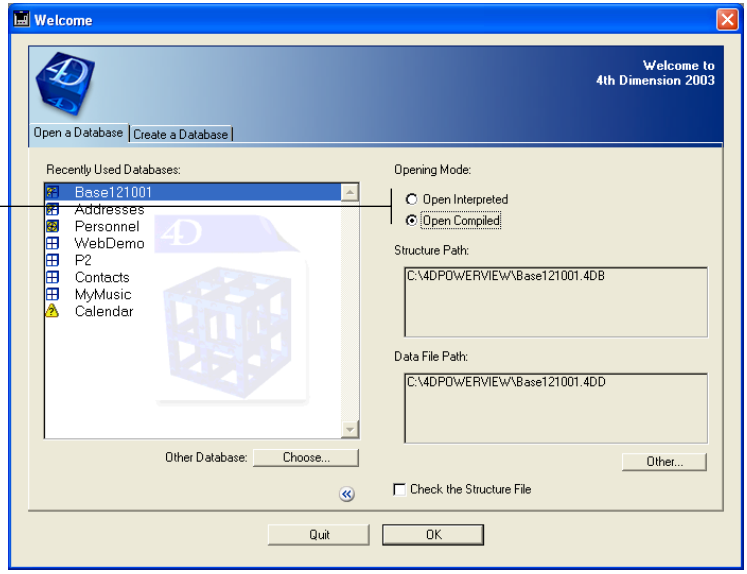
Navigation between interpreted and compiled mode

In 4th Dimension, a structure file may contain both interpreted and compiled code. At the time of database compilation, the compiled code is integrated into the structure file.

The compiled database can therefore function either in *interpreted mode* or in *compiled mode*. This makes it possible to immediately check the behavior, in compiled mode, of an application in the process of being developed.

When a database contains both interpreted and compiled code, 4th Dimension enables you to choose the mode of execution in two different ways: on startup or during use.

- **startup**: In the database opening dialog box, the options area includes the **Open Interpreted**/**Open Compiled** radio buttons that let you choose the mode in which the database will be started:



Note For more information about this dialog box, refer to the paragraph "Opening options," page 31.

- **During use**: The **Use** menu now includes an additional command: **Run compiled** or **Run interpreted** (toggle command). The command is active if the database has been compiled at least once:

This command allows you to change the mode of execution at any time.

Note If you carry out modifications to the database structure in interpreted mode, you must recompile in order to benefit from them in compiled mode.

Changing mode and database methods

When you change from one mode to another, 4th Dimension closes the current mode and opens the new one. This amounts to quitting and reopening the application.
As a result, during a change of mode, 4th Dimension carries out any database methods that may have been defined, in the following order:

- On Exit database method
- On Startup database method

4D Server Changing a 4D Client machine from one mode to another does not modify the session of other connected client machines.

15 Building Final Applications

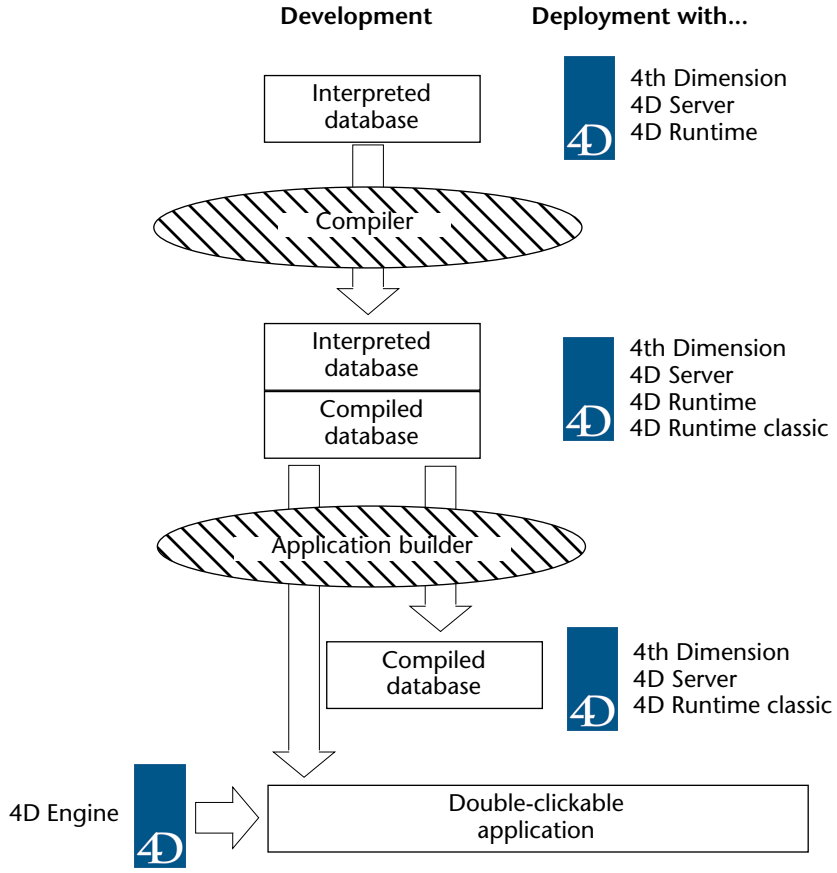
4th Dimension includes a final application builder. This builder simplifies the finalization and deployment process for 4th Dimension compiled applications. It automatically handles the specific features of different operating systems, in particular the building of software packages under MacOS.

The application builder allows you to:

- build a compiled database, without interpreted code,
- build a stand-alone, double-clickable application, i.e., merged with 4D Engine, the 4D database engine.

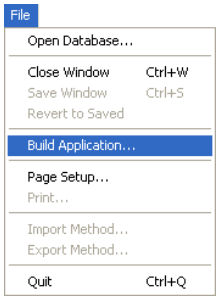
4D Server Application building is available only in the 4th Dimension single-user version.

Deployment possibilities for 4th Dimension databases are summarized in the following diagram:



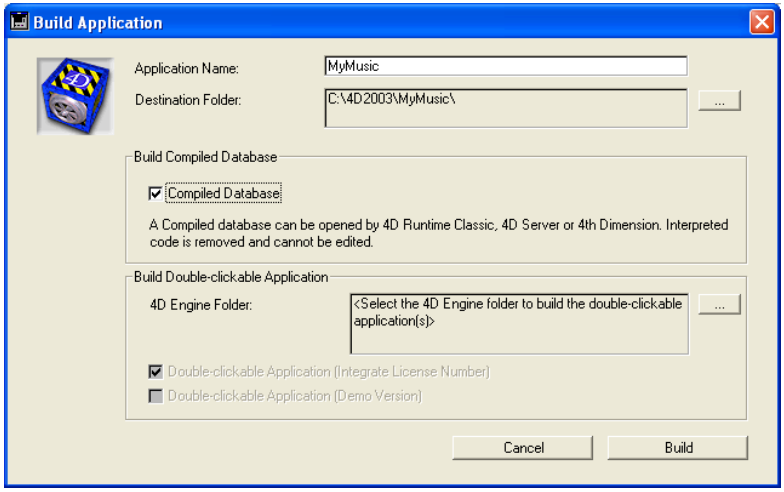
Application builder

Application building is carried out using the “Build Application” window. To display this window, select the **Build application...** command in the **File** menu of 4th Dimension :



Note Building can only be carried out once the database is compiled. If you select this command without having previously compiled the database, or if the compiled code does not correspond to the interpreted code, a warning dialog box appears indicating that the database must be (re)compiled.

The application building window appears as follows:



This window enables you to define:

- the name and location of the files to be generated,
- the type(s) of compiled database(s) to be built:
  - compiled structure file,
  - and/or double-clickable application (with or without an integrated license).
- the location of the 4D Engine file to be used when building a double-clickable application.

During building, 4th Dimension automatically creates the various folders intended to receive the generated files. No matter which option is chosen, the current database is not modified: copies are generated on your disk.

When the compiled structure files comprise both Windows and MacOS code[1], the corresponding versions of the compiled databases will be generated.
On the other hand, a double-clickable application is intended only for the current platform. To build applications available under both MacOS and Windows, it is necessary to use both a MacOS and a Windows version of 4th Dimension.

Once you have configured the various options of this window, click the **Build** button in order to generate the desired files in the location indicated.

When you click on **Build**, the parameters defined in the window (if they are valid) are stored in the database structure file. During subsequent use of this window, the parameters will be kept by default; simply click on the **Build** button to regenerate the files.

1. This option is defined in the application preferences (see the paragraph "Setting Page," page 137).

Definition of the application name and destination

The upper part of the application building window allows you to define the name and location of the files to be generated.

The “Application Name” area contains, by default, the name of the database structure file. This name will be used for the generated files (compiled database and double-clickable application). 4th Dimension will automatically add the necessary suffixes (.4dc, .exe...) according to the type of application built.

When you keep the default name, the name of the application reflects any modification made to the database structure file name.
If you modify the name of the application, the new name is used by default for each new build of the current database. The entered name must therefore NOT have an extension. In addition, it must not contain any characters forbidden by the operating system (such as “.?!” under Windows, “:” under MacOS, etc.).

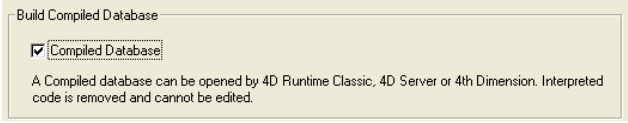
The “Destination Folder” area is used to indicate the location where the generated items will be placed. By default, the folder containing the database structure file is selected. It is useful to keep this default location since, in this case, the destination folder and, if necessary, the 4D Engine folder are stored in relation to the structure file. It is then possible to recopy the folder from one platform to another without having to redefine the pathnames. In addition, you can move or rename the destination folder without holding up the operation of the dialog box.

To modify the destination folder, click on the selection button [...] located to the right of the display area. A folder selection dialog box appears enabling you to indicate the new destination folder. Once this dialog box is validated, the complete pathname of the folder is displayed. The new location will be used by default for each subsequent build of the current database.

At the time of building, 4th Dimension will automatically create one or several intermediary folders (entitled “Compiled database,” “Final application” or “Demo application” according to the type of build requested) in the specified location. This avoids the risk of accidentally deleting files with the same name and enables several types of builds to be performed simultaneously.

Building a compiled database

This option of the application building window is used to build a database containing only compiled code.

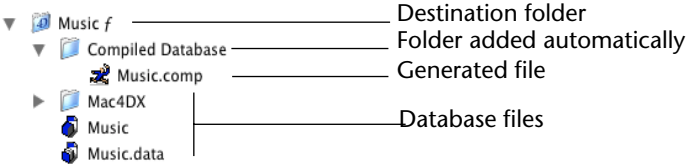


For instance, if you have defined “MyDatabase” in the “Application Name” area, after the build is carried out you obtain:

- under Windows, two files (MyDatabase.4dc and MyDatabase.rsr),
- under MacOS, one file (MyDatabase.comp).

The .4dc file (under Windows) or .comp file (under MacOS) can be opened by 4th Dimension, 4D Server or 4D Runtime Classic. It is no longer possible to access the Design mode of this database.

During the build, 4th Dimension deletes the interpreted code of the structure file. The compiled database is placed in a subfolder named **Compiled Database**. This subfolder is itself created in the specified “Destination Folder.”



Note For more information about the name and pathname of the generated files, refer to the paragraph “Definition of the application name and destination,” page 715.

Prior to rebuilding a compiled database, 4th Dimension erases the previous contents of the "Compiled database" folder. Consequently, you must move any compiled versions and/or additional items that you want to keep beforehand.

Building a double-clickable application

4th Dimension enables you to build a double-clickable application directly from your database. You just need to have 4D Engine, the 4D database engine, and an appropriate license.

The principle consists of merging a compiled structure file with 4D Engine.

Double-clickable (.exe) versions of your 4D compiled databases can be created directly in 4th Dimension 2003 using the "Build Double-clickable Application" function.
Under MacOS, this function handles the creation of software packages. In addition, there is the possibility of building "demo versions" (i.e., not including license numbers) of your applications.

Selection of the 4D Engine folder

To be able to build a double-clickable application, you must first indicate the location of the 4D Engine folder. The options for building double-clickable applications are grayed out if no folder has been indicated in the corresponding area, or if the folder indicated does not contain a valid 4D Engine.

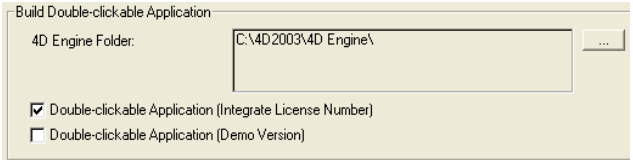
You must select the folder containing the 4D Engine file as well as (under Windows) its associated files.

- Under Windows, the folder contains 4DEngine.4DE, 4DEngine.RSR, ASINTPPC.DLL, and ASIPORT.RSR files, the 4D Extensions folder, as well as the asifont.map and ASIFONT.FON files. These items must be placed at the same level as the selected folder.
- Under MacOS, 4D Engine is provided in the form of a software package containing various generic files.

To select the 4D Engine folder, click on the [...] button located to the right of the "4D Engine Folder" display area.

- Under Windows, a folder selection dialog box appears in order for you to indicate the folder containing the 4D Engine files.
- Under MacOS, a standard file opening dialog box appears, enabling you to select the 4D Engine software package.

Once the folder is selected, its complete pathname is displayed and, if it actually contains 4D Engine, the options for building a double-clickable application are activated.



Licensed version and demo version

4th Dimension enables you to build two types of double-clickable applications:

- **licensed version**: To be able to distribute 4th Dimension executables, you must have the appropriate deployment licenses for 4th Dimension and any plug-ins used in the database.
  These licenses are entered in the license number management dialog box of 4th Dimension (accessible via the 4th Dimension **Help** menu). When this option is checked, 4th Dimension builds a double-clickable application integrating the deployment license numbers present on the machine. The application is thus completely operational.
- **demo version**: When this option is checked, 4th Dimension does not integrate any license number at the time of building. Consequently, the applications will only operate in "demo" mode (limited number of tables and records).

You can select both options simultaneously. Each type of application will be built in a specific subfolder of the destination folder:

- Licensed versions are built in a subfolder entitled **Final application**.
- Demo versions are built in a subfolder entitled **Demo application**.

For more information about the name and pathname of the generated files, refer to the paragraph "Definition of the application name and destination," page 715.

When you click on the **Build** button, 4th Dimension displays a progress thermometer indicating the different phases being performed. If they are valid, the new parameters are saved.

Note If you click on the **Cancel** button, or if an error occurs during the process, the files being generated are deleted and a warning dialog box informs you of the cause of the interruption.

Generated files

At the end of the build, the following files should be found in the **Final application** or **Demo application** subfolder of the destination folder:

- **Under Windows**
  - DatabaseName.EXE which is your executable,
  - DatabaseName.4DC and DatabaseName.RSR which make up your compiled structure,
  - ASINTPPC.DLL, ASIPORT.RSR files, 4D Extensions folder and asi-font.map and ASIFONT.FON files,
  - Additional items possibly added in the 4D Engine folder (see the paragraph "Customizing the 4D Engine folder," page 720).

  All these items must be kept in the same folder in order for the executable to operate.
- **Under MacOS**
  - A software package entitled DatabaseName.app containing your application and all the items necessary for its operation.

Note Under MacOS, the Application file command of the 4D language now returns the pathname of the ApplicationName file (located in the Contents:MacOSClassic or Contents:MacOS folder of the software package) and not that of the .comp file (Contents:Resources folder of the software package).

- **On both platforms**, the list of license numbers that have been integrated into the licensed version is registered, if necessary, in a "licencelog.txt" file that is automatically placed in the **Final application** folder.

Rebuilding a double-clickable application

Prior to rebuilding a double-clickable application, 4th Dimension erases the contents of the "Final application" and/or "Demo application" folders. Therefore, you must move any versions that you want to keep.

Customizing the 4D Engine folder

When building a double-clickable application, 4th Dimension copies the contents of the 4D Engine folder into the **Final application** and/or **Demo application** subfolder of the destination folder. You are then able to customize the contents of the original 4D Engine folder according to your needs.

You can, for instance:

- customize 4D Engine operation using the 4D Customizer Plus utility;
- install a 4D Engine version corresponding to a specific language;
- add a Mac4DX and/or Win4DX folder containing the plug-ins necessary for the operation of the application;
- customize the contents of the 4D Extensions folder.

Note Under MacOS, 4D Engine is provided in the form of a software package. In order to modify it, you must first display its contents (**Control+click** on the icon). The Win4DX and Mac4DX folders must be placed at the same level as the Resources folder.

Customizing the application icon

4th Dimension associates a default icon with double-clickable applications. However, you can customize the icon for each application.

Under MacOS 9

In order to customize your compiled application under MacOS, you must modify the signature and icon of the newly-created application. To do this, use a resource editor of the ResEdit® type.

To change the signature of your application, two operations are necessary:

- First, open the BNDL type resource. Replace the 4 characters of the application signature by the 4 characters that you have chosen to be the new application signature. Make sure that this signature is unique in order to avoid any risk of confusion during start-up of the application. If you use the signature of another program, the Finder may switch the different icons.
- Second, open the SIG* resource and copy the same 4 characters you just assigned to the BNDL resource.
  If your resource editor has not automatically done so, create a resource of the same type as that of the signature you gave to your application.

This resource, whose ID number will be 0, will contain, for instance, the name of your application.

You can then customize the icon of your application. This icon is located in the icl8, icl4, ICN#, ics#, ics4 and ics8 resources.

1 **In each of these resources, open the icon with the ID number 128.**

2 **Using the resource editor, modify this icon as desired.**

If you stop there, the application icon is customized, but the files that will be created (data file, export files, etc.) will have standard 4th Dimension icons. These icons, whose ID numbers are greater than 128, can also be customized using the same procedure.

3 **Open the icl4 resource again.**

Each of the icons in the list corresponds to a type of file generated by 4th Dimension: .data, ASCII export, etc.

4 **Modify these icons as desired.**

This modification will only be made for files created with the compiled and customized database; previous files will keep their original icons.

Notes

- If you want to market your application, it will first be necessary to conform to the conditions set by Apple. In this way, you can be sure that your signature is unique and will not produce any icon conflicts.
- It is possible that modifications will not be performed in real time. In this case, you must rebuild your desktop before you will be able to display them. To do this, restart your Macintosh while holding down the **Command** and **Option** keys.

Under MacOS X

When building a double-clickable application, 4th Dimension handles the customizing of the icon.
In order to do this, you must carry out the following operation prior to building the application file:

1 **Create an icon file (icns type) and place it next to the interpreted structure file.**

Note Apple, Inc. provides a specific tool for building icns icon files.

Your icon file must have the same name as the interpreted structure file and include the .icns extension.

4th Dimension automatically takes this file into account when building the double-clickable application (the .icns file is renamed ApplicationName.icns and recopied into the Resources folder; the *CFBundleFileIcon* entry of the "info.plist" file is updated).

Under Windows

When building a double-clickable application, 4th Dimension handles the customizing of its icon.

In order to do this, you must carry out the following operation prior to generating the application file:

1 **Create an icon file (.ico extension) and place it next to the interpreted structure file.**

Your icon file must have the same name as the interpreted structure file and include the .ico extension.

4th Dimension automatically takes this file into account when building the double-clickable application.

16 Publishing or Using Web Services

Presentation

4th Dimension includes features that enable the publishing or use of Web Services within your database.

What are Web Services?

A Web Service is a set of functions grouped together as an entity and published on a network. These functions can be called and used by any application compatible with Web Services and connected to the same network. Naturally, Web Services are intended to be used to their fullest in the context of publishing on the Internet.

Web Services can carry out all types of tasks, such as supervising the routing of packages at a transporter's, e-commerce, monitoring market values, etc.

The program publishing the services is called the "server." Any application compatible with Web Services can therefore use one or more of these functions; this is the "client" program.
The advantage of Web Services is their interoperability with different information systems: it is not necessary for the server and client programs to be mutually compatible in order for the system to work. From the client application point of view, a Web Service is a "black box": values are sent to it and other values resulting from processing are returned.

The Web Services proposed by the server can be either public or private. There are a great number of public Web Services on the Internet that any application can solicit free of charge.

Maintained by the W3C (World Wide Web Consortium, the regulating authority of the Internet) and major firms of the computer industry, Web Services represent a reliable, lasting and upgradable connectivity solution.

Operation of Web Services —main definitions

Web Services transit essentially using the HTTP transport protocol.

SOAP: Web Services use an "open" high-level communication protocol named SOAP (Simple Object Access Protocol). This protocol is based entirely on the XML language, both at the level of the message structure (envelope) and that of the exchanged data. The operation of this protocol is defined by the RFCs (Request for Comment, documents standardizing the various aspects of the Internet), which guarantees it widespread compatibility.

The operating principle of a Web Service is as follows: a Web Service client sends a request in XML to the server via the SOAP protocol. The server analyzes the request, carries out the requested operation, and returns its response using the same protocol and language.

WSDL: The servers of Web Services generally publish a WSDL (Web Service Description Language) in order to define access specifications for the services being offered. The WSDL enables servers of Web Services to publish the "operating instructions" of the services offered (URLs, lists of methods, parameters, etc.) and comes in the form of an XML file, generally created by the server application itself. This file is not mandatory.

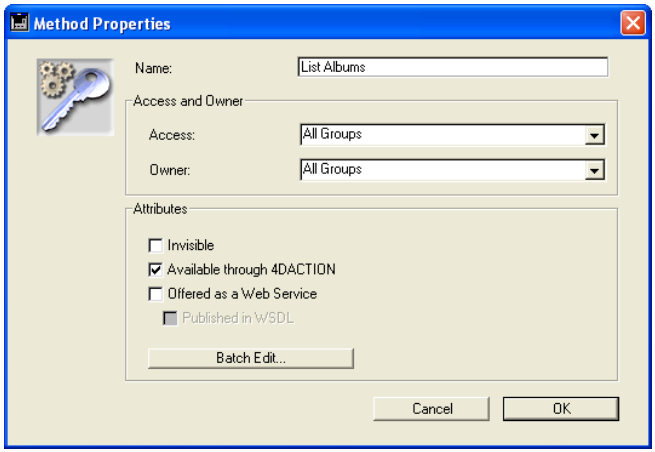
UDDI: The UDDI (Universal Description Discovery and Integration) is a worldwide database that inventories all the public Web Services. Note that it is not obligatory to make a Web Service public and in most cases this will not be necessary.
For more information about the UDDI, please refer to the following address: http://www.uddi.org.

Integration of Web Services in 4D

4th Dimension can be used as a Web Services server and/or client. Integration of Web Services into 4th Dimension is simple and secure: several configurations enable precise monitoring of publication and subscription conditions.

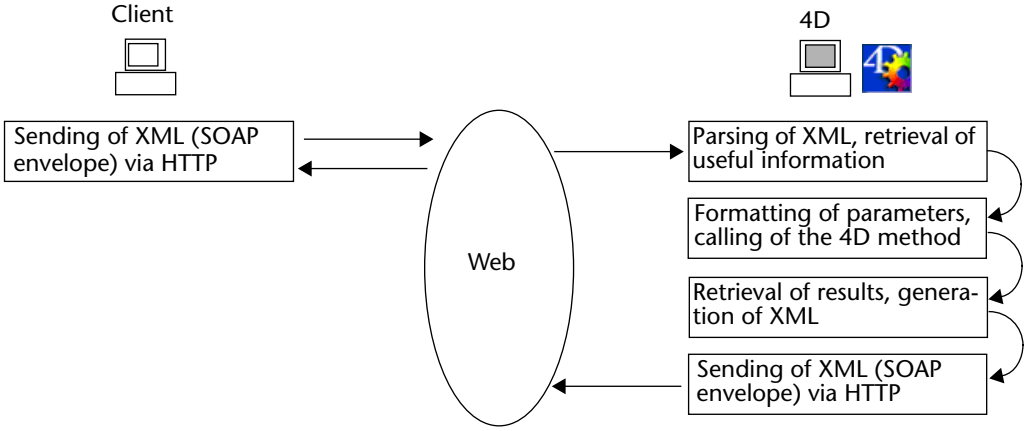
4D as a Web Services server

You can decide to publish any project method as a Web Service, without any major modification. Publication is a method property:



The 4D Web server automatically handles the management of the service as well as the publication and maintenance of the WSDL file. Parsing the XML content of requests, parameter formatting, sending of results, etc., are performed by 4th Dimension without any specific programming being necessary.

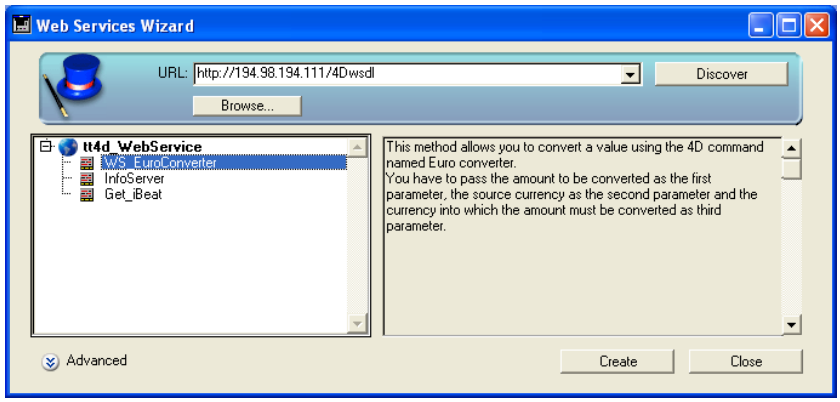
Processing of SOAP requests by 4D



However, if you want to customize request processing, you can use the specific 4th Dimension language commands — refer to the "Web Services (Server)" section in the *4th Dimension Language Reference* manual.

4D as a Web Services client

Your databases can use any type of Web Service offered on the Internet or on your network. Most of the time, the Web Services Wizard will enable you to use any Web Service instantly, with a minimum of programming :



Using a Web Service in 4D consists in sending requests over the network and retrieving a response. “Proxy” methods are in charge of these operations. The creation of proxy methods for calling Web Services is entirely automatic and can be performed without programming. Simply call these methods in your code.

It is possible to customize these methods using 4th Dimension language commands, in the same way as for the server part.

Security of Web Services

Web Services published by 4D inherit security mechanisms set up for the 4D Web server. Web Services requests thus benefit from the same configurations as conventional Web requests: passwords, On Web Authentication and On Web Connection database methods, use of the SSL protocol, etc.
In addition, specific configurations (for example, Get SOAP info and Is SOAP request commands) allow precise control of Web Service publication.

On the client side, connection to Web Services servers can be carried out in secure mode using SSL. A specific command also allows connection to servers requiring authentication.

Compatibility of RPC, DOC and complex types

The communication layer of Web Services (ensuring transport, calling of services and security of exchanges) can operate in two different modes: **RPC** (Remote Procedure Call) mode and **DOC** (Message/Document) mode. The difference between these modes lies at the building level of requests and responses for the server and client.

The compatibility of 4th Dimension with these modes is as follows:

- Server side, 4th Dimension publishes Web Services in RPC mode only.
- Client side, 4th Dimension supports both RPC and DOC mode.

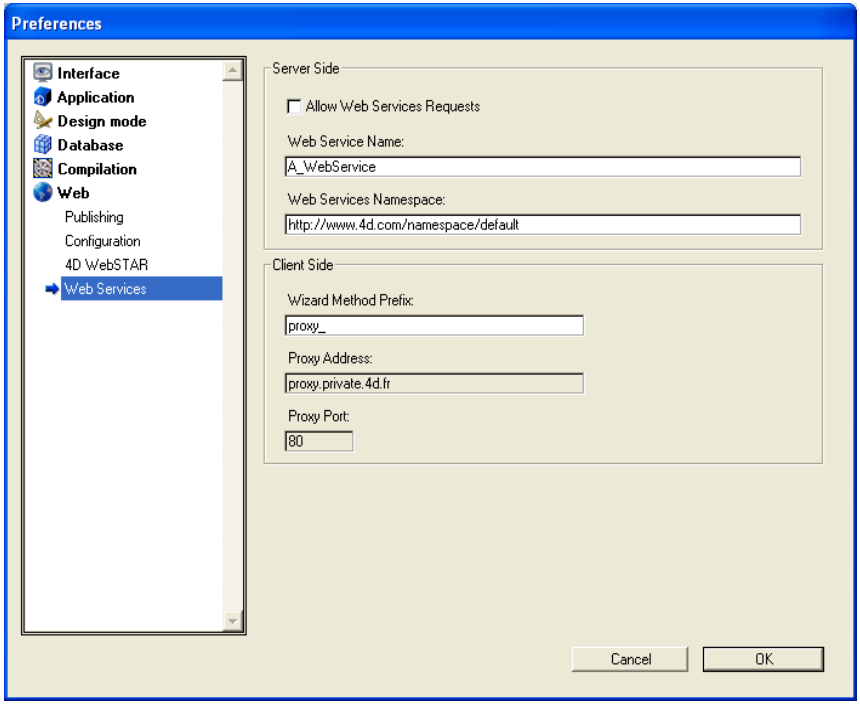
  Two different types of XML data are exchanged via the SOAP protocol: **simple** types and **complex** types. The data of Web Services published in RPC mode can be of either type. Conversely, the data of Web Services published in DOC mode are systematically of the complex type.

- On the server side, 4th Dimension only publishes simple data types, with the exception of data arrays.
- On the client side, 4th Dimension supports Web Services using simple data types (RPC mode) and Web Services using complex data types (RPC or DOC mode).
  Note that except for arrays[1], complex XML data types cannot be used directly in a 4D database and require specific processing. This additional processing is partially handled by the Client Web Services Wizard, but also requires programming. For more information about this point, refer to the paragraph “Processing complex types,” page 745.

1. Data arrays are complex XML data types but are not considered as such by 4D and are handled automatically.

Configuration

The **Web Services** page of the database Preferences (“Web” theme) can be used to define the general parameters concerning the publication and use of Web Services:



The action of these parameters are described in the following pages. For a more detailed description of each option, refer to the paragraph “Web Services Page,” page 150.

Publishing a Web Service with 4th Dimension

Publication of a Web Service in 4th Dimension is generally carried out in three stages:

1 **Creation of the method to be published,**

2 **Configuration of the publication (WSDL),**

3 **Publication.**

Additional customizing stages can be defined, but they are not mandatory.

Creating a Web Service method

You can create any type of project method intended for publication as a Web Service. The method must accept parameters and return a result. It is imperative that these parameters be declared in the method header using commands of the "Compiler" theme.

By default, 4th Dimension formats the parameters necessary for the operation of methods when published as Web Services. You can, however, modify these parameters using the SOAP DECLARATION command.

4th Dimension automatically takes care of decoding and encoding the data received and sent via SOAP.

Warning: The names of methods are used as XML tags in SOAP requests. In conformity with the XML standard for tag names, the names of methods published as Web Services must NOT contain spaces or extended characters. Only the following Latin characters may be used: ([A-Za-z0-9._] | '-')*.

To define and monitor the development of a method published as a Web Service, you must use the commands of the "Web Services" theme— refer to the "Web Services (Server)" section in the *4th Dimension Language Reference* manual.

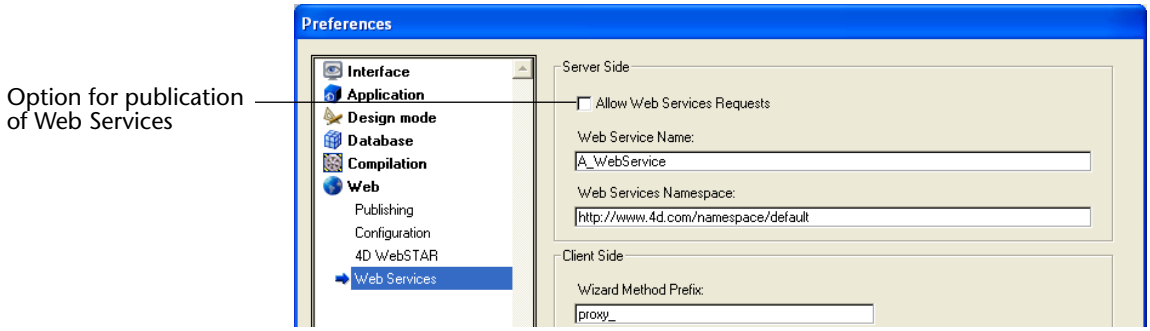
Publication of methods

To be able to publish one or more methods of your database as Web Services, the three following conditions must be met.

Note The publication of Web Services requires that the machine used as SOAP server (4D Server, 4D Client or single-user 4th Dimension) have a 4D Web Server license.

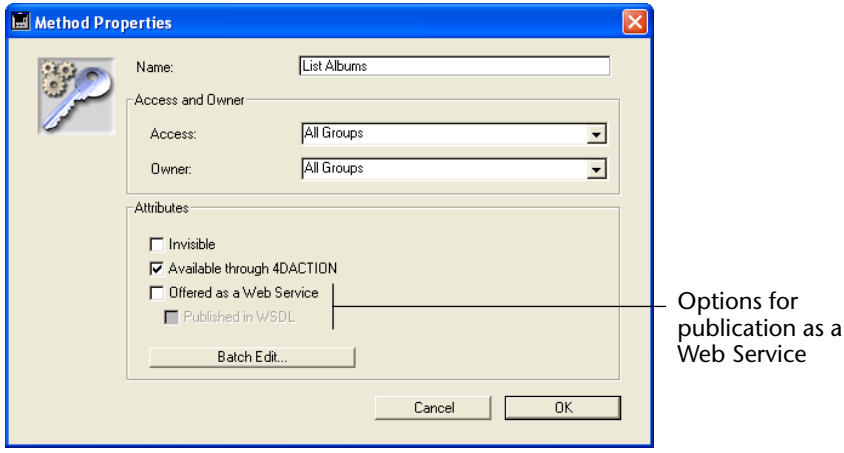
- The 4D Web server must be launched (to check that it is, pull down the **Web Server** menu in User mode: the **Start Web Server** command should be deactivated).
- The **Allow Web Services Requests** option on the "Web Services" page in the Preferences dialog box of the application must be checked. If it

is not, 4th Dimension refuses SOAP requests and does not generate a WSDL.



When this option is checked, 4th Dimension creates the WSDL file (see the paragraph "Generation of the WSDL," page 731).

- Each method to be published must be **Offered as a Web Service**. This configuration is carried out using an option located in the Method Properties window:

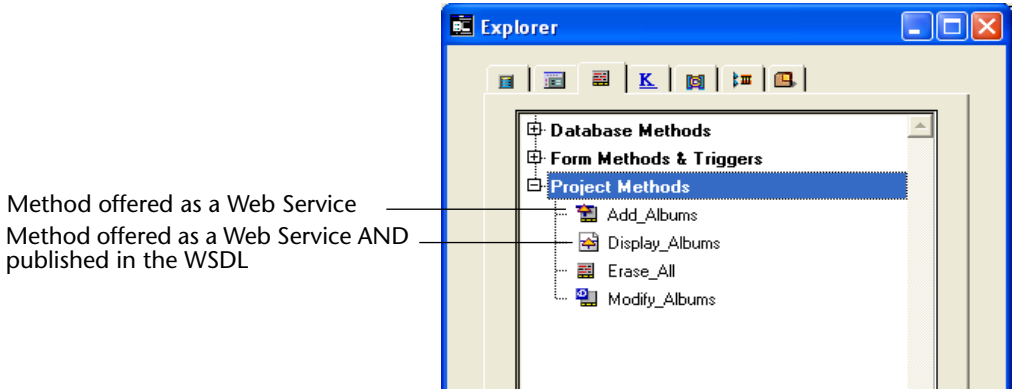


Note For more information about this window, refer to the paragraph "Defining the properties of project methods," page 541.

When the **Offered as a Web Service** option is checked, the corresponding method can be called as a Web Service via a SOAP request.

Note If the **Published in WSDL** option is also checked, the method will appear in the WSDL of the server (see the following paragraph).

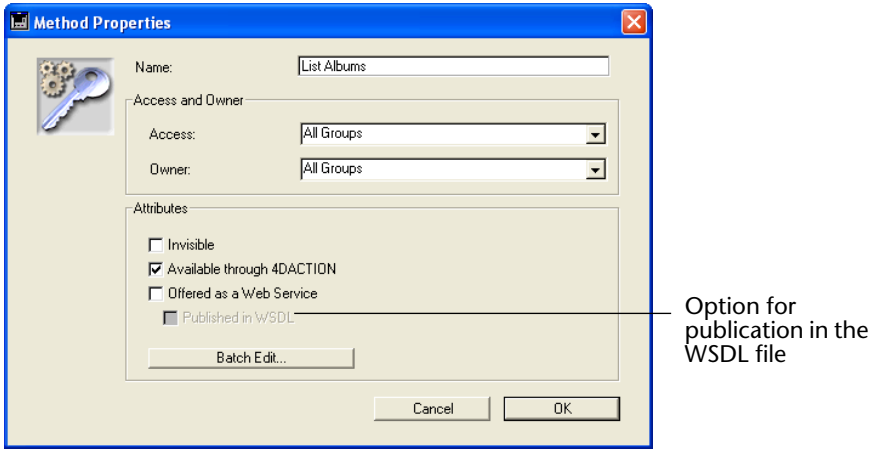
In the 4D Explorer, a specific icon indicates the methods offered as Web Services (as well as those published in the WSDL file) :



Generation of the WSDL

In 4th Dimension, the WSDL corresponds to a single Web Service. It defines the methods and their parameters and can be consulted at a specific location. In 4D, the WSDL is not an actual “file” (it only exists in memory and is not written to disk); it is a URL always named **4DWSDL** and located at the root of the Web server. For example, if the address of your Web server is http://www.myserver.com, you could consult the WSDL at the following URL: http://www.myserver.com/4DWSDL.

When Web Services requests are allowed, 4th Dimension automatically and dynamically generates the WSDL of the 4D Web server if at least one method has the **Published in WSDL** option checked in the Method Properties window:



By default, this option is not checked.

This document describes, in XML language, the syntax and information needed for calling the 4D method (Method name, URL, parameters, etc.).

Example of the WSDL file displayed in a Web browser

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- this WSDL file was automatically generated by 4D -->
- <definitions name="tt4d_WebService" targetNamespace="http://www.4d.com/namespace/default"
  xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:SOAP-
  ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.4d.com/namespace/default" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance">
 - <types>
  - <schema targetNamespace="http://www.4d.com/namespace/default" xmlns="http://www.w3.org/2001/XMLSchema"
      xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
    - <complexType name="ArrayOfstring" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
      - <complexContent>
        - <restriction base="SOAP-ENC:Array">
            <attribute ref="SOAP-ENC:arrayType" wsdl:arrayType="xsd:string[]" />
          </restriction>
        </complexContent>
      </complexType>
    </schema>
  </types>
 - <message name="WS_EuroConverterRequest">
    <part name="Currency_in" type="xsd:float" />
    <part name="From" type="xsd:string" />
    <part name="To" type="xsd:string" />
  </message>
 - <message name="InfoServerRequest">
    <part name="Selector" type="xsd:int" />
  </message>
```

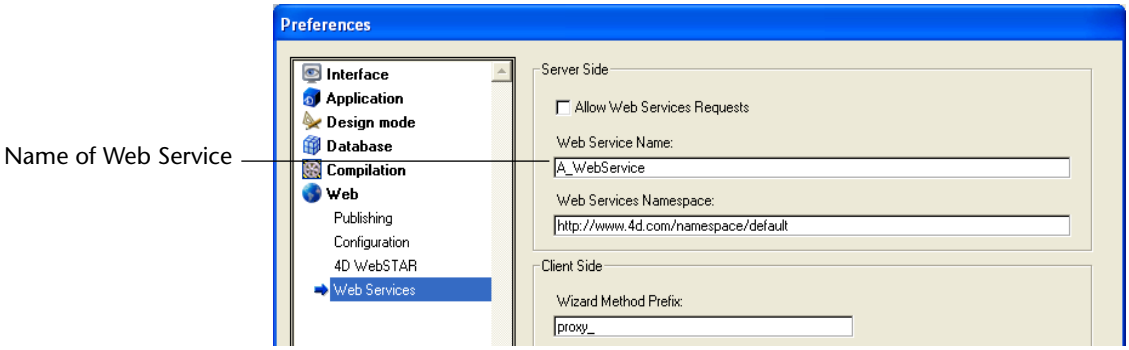
To add or remove a method in the WSDL, simply check or uncheck the corresponding option in the Method Properties window. 4th Dimension will immediately update the WSDL contents.

Note In the 4th Dimension Explorer, a specific icon distinguishes methods that are published in the WSDL (see the previous paragraph).

Customizing a Web Service name

Each Web Service published on the Internet has a name. This name is used to differentiate the services both at the SOAP server level (when the server publishes several different Web Services), as well as in the Web Services directories.

By default, 4th Dimension uses the name A_WebService. This parameter can be modified on the **Web Services** page of the "Web" theme in the database Preferences:

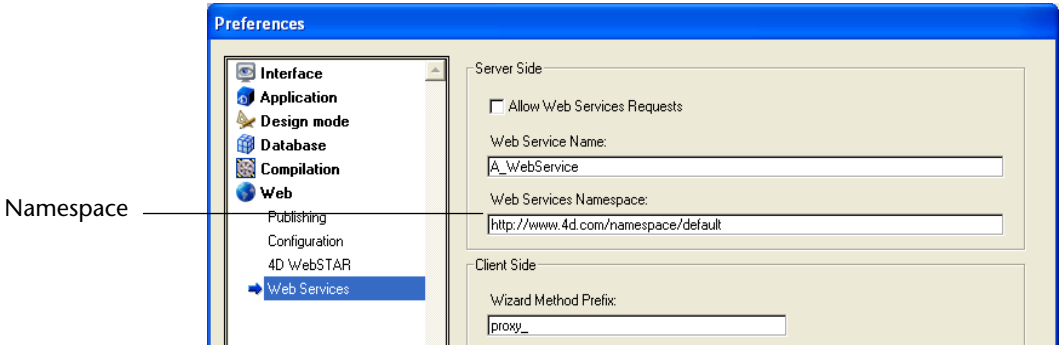


Warning: In conformity with the XML standard for tag names, the character strings used must not contain spaces or extended characters. Only the following Latin characters may be used: ([A-Za-z0-9._] | '-')*

Customizing a Namespace

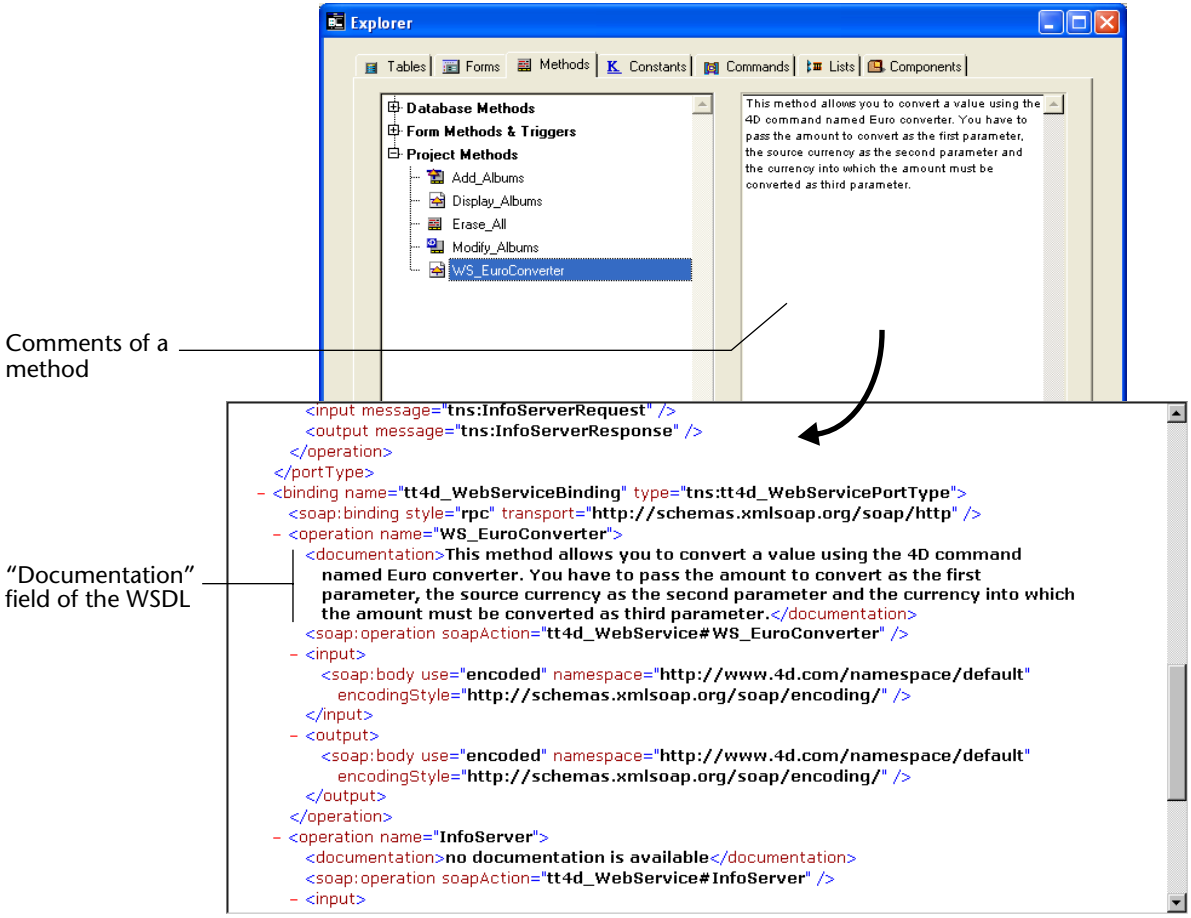
Each Web Service published on the Internet must be unique. The uniqueness of the names of Web Services is ensured using XML namespaces. A namespace is an arbitrary character string used to identify a set of XML tags in a unique way. Typically, the namespace begins with the URL of the company (http://mycompany.com/mynamespace). In this case, it is not indispensable to have anything in particular at the defined URL; what matters is that the character string used is unique.

By default, 4th Dimension uses the following namespace: http://www.4d.com/namespace/default. This parameter can be modified on the **Web Services** page of the "Web" theme in the database Preferences:



Adding comments to published methods

Any comments associated with methods offered as Web Services and published in the WSDL automatically appear in this file as a "documentation" field.



This mechanism is used to describe or document the published methods. The interpretation and handling of this field will depend on the implementation of the client Web Service.

Accessing a Web Service published by 4th Dimension

Once your Web Service has been published by 4th Dimension, any client application that supports Web Services can connect to it. The access mode and the processing of the information exchanged with the Web Service server will depend on the client application used for the operation.

All information needed for the use of a Web Service (such as the URL of the service, the parameters to be used, etc.) are published in the WSDL of 4th Dimension. In principle, the use of a Web Service should thus begin with the reading of the WSDL of the SOAP server in order to retrieve this information. In 4th Dimension, the URL of the WSDL is always http://ServerAddress/4DWSDL.

However, this step is not mandatory. Connection to the SOAP server can be carried out directly.

Here is a list of the values needed to establish SOAP requests, as well as their method of definition:

- **Access URL to a Web Service published by 4D**
  http://ServerAddress/4DSOAP/ (not customizable)
- **Web Service name**
  By default: A_WebService.
  Customizable value (see the paragraph "Customizing a Web Service name," page 732).
- **Name of published method**
  Name of the 4D project method defined by the developer (see the paragraph "Creating a Web Service method," page 729).
- **Method parameters**
  The parameters must be declared in the method (defined by the developer).
  Default SOAP names: FourD_arg0, FourD_arg1... FourD_argn.
  Customizable names using the SOAP DECLARATION command.
- **Namespace**
  By default: http://www.4d.com/namespace/default.
  Customizable value (see the paragraph "Customizing a Namespace," page 733).
- **Contents of SOAP Action field**
  ServiceName#MethodName (not customizable).

Subscribing to a Web Service in 4th Dimension

4th Dimension allows you to subscribe to Web Services; in other words, to call external Web Services from within your databases.

Using Web Services available on the Internet, you can easily add numerous additional functions to your databases, such as access to stock market information, package delivery follow-up, execution of complex calculations, etc. The multitude of Web Services available on the Internet can fulfill almost every need.

You can also subscribe to Web Services that you have published yourself in other databases and in this way let various 4th Dimension databases communicate among themselves.

Principles

Any 4th Dimension database can subscribe to a Web Service; it simply needs to be connected to the Internet.

Generally, to be able to call a Web Service, you must follow the steps described below:

1 **Retrieve the URL of the Web Service to which you want to subscribe.**
To do this, you can use Web sites that inventory Web Services published on the Internet (for example www.xmethods.net) or directories such as the UDDI. In most cases, you must obtain the URL of the WSDL file for the Web Service.

Note 4th Dimension can use Web Services published in RPC or DOC mode (see the paragraph "Compatibility of RPC, DOC and complex types," page 727).

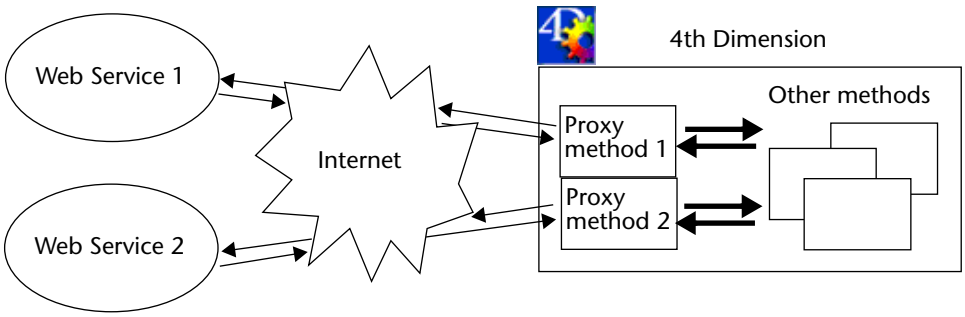
2 **Using the Web Services Wizard, parse the contents of the WSDL of the Web Service to be used and generate the corresponding *proxy method*.**
The proxy method is the local project method in charge of interrogating the Web Service and retrieving the returned values. This step is described in the paragraph "Using the Web Services Wizard," page 737.

Note
- It is possible to create proxy methods in the Web Services Wizard without using a WSDL file (simply enter the parameters to be used manually).
- It is also possible to create proxy methods in the Method editor, without using the Web Services Wizard (advanced users).

3 **In the code of your database, call the proxy method each time that you need it by passing the appropriate parameters to it.**
This step is described in the paragraph “Calling a proxy method,” page 745.

The proxy method handles the connection to the Web Service:

Principle of operation of 4D as a client Web Service



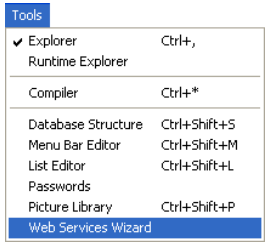
Using the Web Services Wizard

The subscription to a Web Service from a 4th Dimension application is handled entirely by the Web Services Wizard. This wizard automatically carries out:

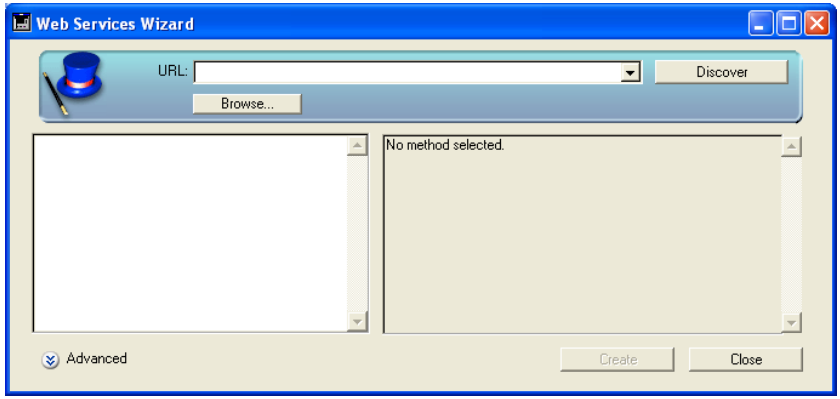
- parsing of WSDL files for the Web Services to be used,
- definition of the parameters for the proxy methods to be created,
- creation of proxy methods.

Wizard window

To open the Web Services Wizard window, choose the **Web Services Wizard** command in the **Tools** menu of 4th Dimension:



The Wizard window appears:



This window includes three areas:

- the “URL:” area allows you to enter or select the URL of the WSDL file for the chosen Web Service. This area is a combo box that stores the previously-entered values in the form of a drop-down list.
- the central area displays the results of parsing the WSDL file contents: names of services and published methods.
- the lower area (“Advanced” parameters, hidden by default) displays the parameters of the method selected in the central area.

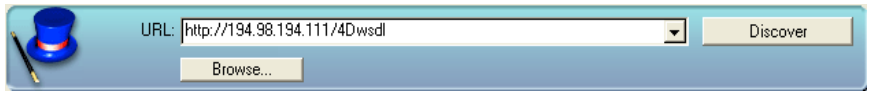
The **Discover** button triggers the parsing of the designated WSDL file and the filling in of the information areas.
The **Browse...** button displays a standard file opening dialog box, allowing the selection of a WSDL stored locally. Its pathname, beginning with “file://”, is then displayed in the “URL:” area (it is possible to enter the pathname manually in this area).
The **Create** button is used to generate the proxy method corresponding to the selected Web Service.
The **Close** button recloses the Web Services Wizard dialog box.

Parsing of a WSDL and creation of the proxy method (standard mode)

Typical use of the Web Services Wizard consists of parsing a WSDL file then generating the corresponding proxy method(s). This standard operation is entirely automatic and does not require any programming or any particular know-how on the part of the user.

► To parse a WSDL file and generate the proxy method:

1 In the "URL:" area, enter or paste the URL of the WSDL file for the Web Service that you want to use:



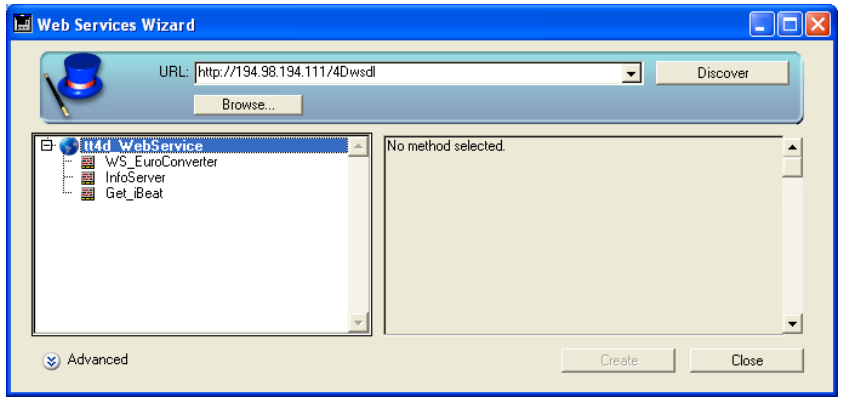
This URL may come, for instance, from a "directory" of Web Services or may have been communicated directly by the server of the Web Service.

You can also specify a local URL, i.e., the address of a WSDL file stored on your hard disk. To do this, click on the **Browse...** button and choose the local WSDL file, or enter its pathname directly in the "URL:" area. The pathname of the local file begins with "file://" then uses the standard system folder separator. You must pass an absolute pathname.

Note In our example, we use a Web Service published by 4th Dimension; however, you can choose any type of Web Service.

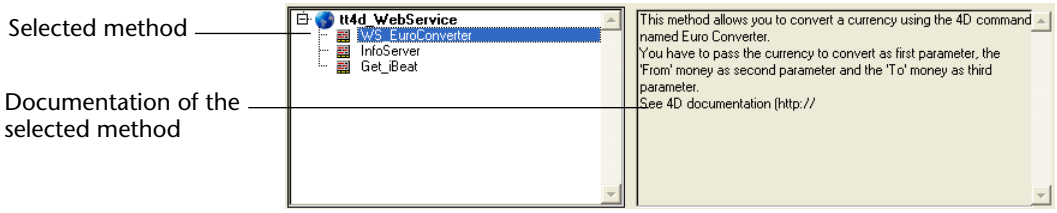
2 Click on the Discovery button in order for 4th Dimension to parse the contents of the WSDL file.

After a few moments, the central area displays the results of file parsing: the name of the Web Service(s) as well as the published methods appear in the form of a hierarchical list.



Note You can display the XML source code of the WSDL file directly in your default Web browser by holding down the **Shift** key when you click on the **Discover** button.

Clicking on a Web Service displays its documentation (if any) on the right-hand side of the window. Otherwise, the indication “No documentation” appears.
Similarly, the documentation (if any) for each method appears when you select its name:

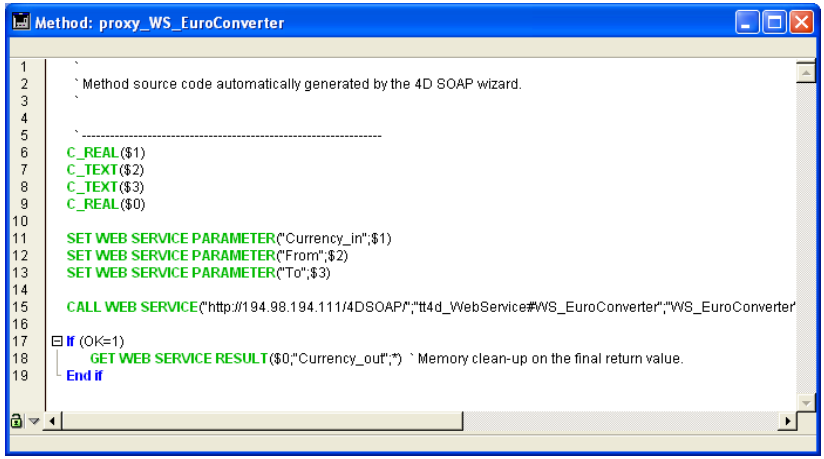


Note If the parsing of the WSDL file reveals the presence of **complex type** parameters, the Wizard displays a yellow flag next to the method concerned. In this case, the integration of the Web Service into your database will require additional programming on your part as well as knowledge of the XML language. For more information, refer to the paragraph “Processing complex types,” page 745.
The presence of complex types does not, however, prevent the Wizard from generating the corresponding proxy method.

GetDefinition

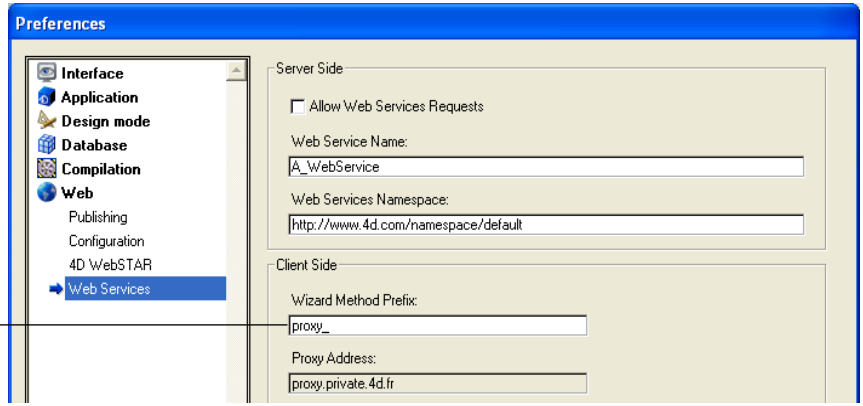
3 Select the Web Service method that you want to use then click on the Create button.

4th Dimension instantly generates the corresponding proxy method and displays it in a window of the Method editor:



The name of the proxy method is defined by the concatenation of the default prefix “proxy_” and the name of the Web Service method.

The default prefix can be modified in the database Preferences on the **Web Services** page:



Default prefix of proxy method names

The name of the proxy method can also be modified after its creation; this does not influence the operation of the method.

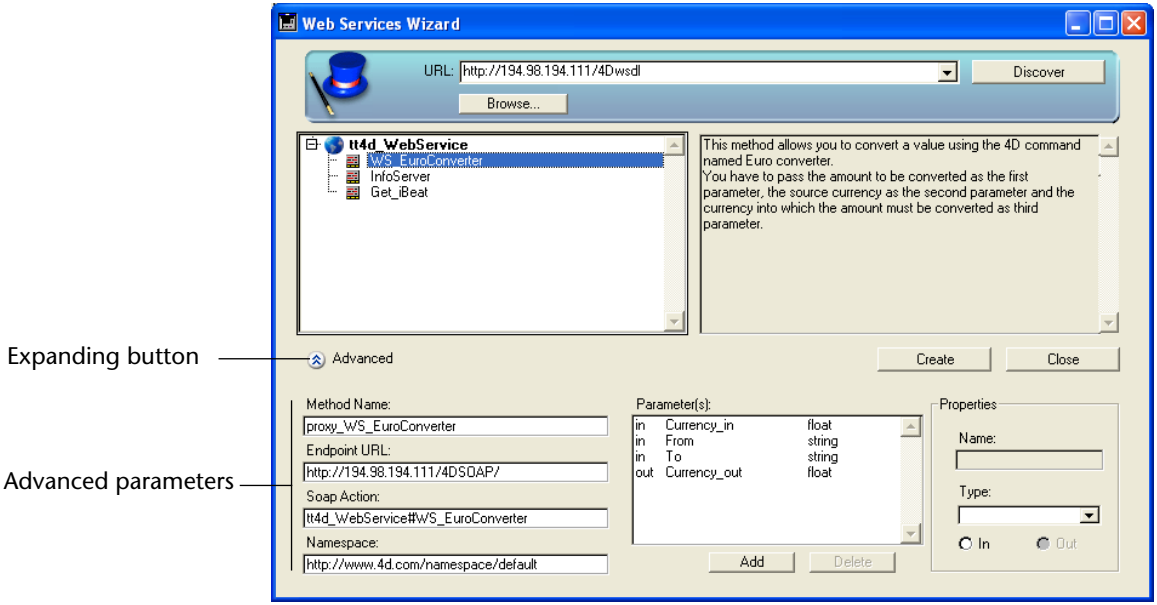
Using advanced parameters

Proxy methods generated by the Web Service Wizard from the parsing of a WSDL file are immediately operational and can be used as is (*standard mode*).

However, you might want to modify the parameters resulting from WSDL parsing. For example, it is possible to rename the proxy method.

You can also use the Web Services Wizard to create a proxy method for which you have manually entered the parameters. In this case, do not use the WSDL parser.
It is not mandatory to enter all the parameters to be able to create a method.
It is even possible to not enter any parameters in order to create a proxy method “template” that you can then fill in using 4th Dimension programming.

In these non-standard modes, you must use the advanced parameters of the Web Services Wizard. To display these parameters, click on the expanding button located at the bottom left of the Wizard window. The advanced parameter fields then appear. If a method is selected, the fields display its current parameters:

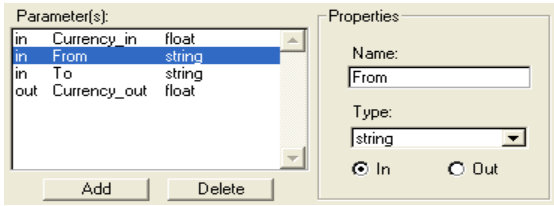


All the parameters are modifiable. Note, however, that modifying parameters stemming from WSDL parsing (except for the method name) must be done with precaution because the operation of the Web Service may be altered as a consequence.

Here is a description of the advanced parameters:

- **Method name**: Name that the Wizard will give to the proxy method to be created. By default, this name is made up of the prefix “proxy_” (modifiable in the Preferences) followed by the name of the selected method. This name can be modified freely (for instance, if it already exists in the database) without this having any influence on the operation of the Web Service.
- **Endpoint URL**: URL to which the proxy method sends the SOAP requests.
- **Soap Action**: Contents of the SOAPAction field. This field generally contains the value “ServiceName#MethodName”.

- **Namespace**: Namespace of the Web Service (for more information, refer to the paragraph “Customizing a Namespace,” page 733).
- **Parameter table**: This table lists the parameters of the published method.



Each row of the table describes a parameter:

- the first column indicates whether the parameter is of the input (“in”) or output (“out”) type. This characteristic is evaluated from the point of view of the proxy method, and not that of the published method.
- the second column indicates the name of the parameter.
- the third column indicates the SOAP type of the parameter. Different SOAP types accepted by 4th Dimension can be displayed in the Type menu located in the Properties area. The Web Services Wizard will be responsible for associating SOAP types with the corresponding 4th Dimension types in the proxy method.
  The following table lists the types of SOAP values accepted and the corresponding 4th Dimension types:

SOAP Type	Corresponding 4D Type
boolean	Boolean
int	Long Integer
time	Time
float	Real
double	Real
date	Date
string	Text
base64Binary	BLOB
ArrayOfBoolean	Boolean array
ArrayOfInt	LongInt array
ArrayOfTime	LongInt array
ArrayOfFloat	Real array
ArrayOfDate	Date array

SOAP Type	Corresponding 4D Type
ArrayOfString	Text array
AsXML[1]	BLOB

1. The AsXML type is not, strictly speaking, a SOAP type, but it is used for supporting complex XML types (see the paragraph "Processing complex types," page 745).

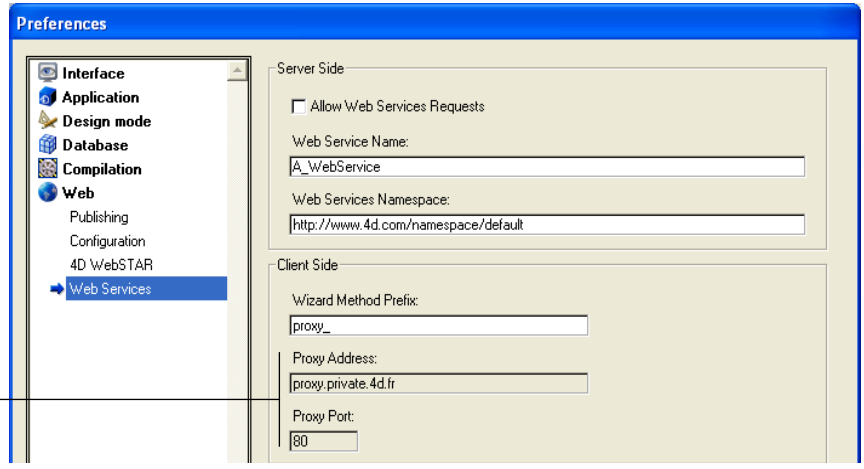
The Properties area displays the characteristics of the parameter selected in the table. The Web Services Wizard allows you to modify the existing parameters or add parameters, for instance if the specified WSDL file is not up to date.

- To modify a parameter, select it and then make your modifications in the Properties area.
- To add a parameter, click on the **Add** button then define its characteristics in the Properties area.
- To delete a parameter, select it in the list then click on the **Delete** button.

Note Modifications made in the advanced parameters will only be taken into account if a proxy method is actually created using the **Create** button.

Display of connection parameters

When you subscribe to a Web Service, 4th Dimension uses the current Internet connection parameters defined for the machine, in particular the Proxy server. You can check these parameters on the **Web Services** page of the "Web" theme in the database Preferences:



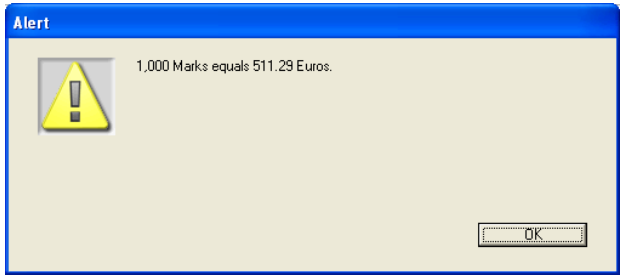
These values are only read by 4th Dimension. If you want to modify them, you must do so via the Internet parameters of the machine.

Calling a proxy method

To call a proxy method in your code, simply write its name and pass the required parameters to it. These parameters are declared in the header area of the proxy method by the Web Services Wizard. In conformity with the standard syntax for passing parameters between methods in 4D, they are named $0, $1, $2... They can be displayed in the advanced parameters of the description of the published method (see the paragraph "Using advanced parameters," page 741) and are sometimes described in its documentation.

For instance, if we go back to the method used on the previous pages (*WS_EuroConverter*), the proxy method could be called in the following manner:

After execution of the method, the following warning is displayed:



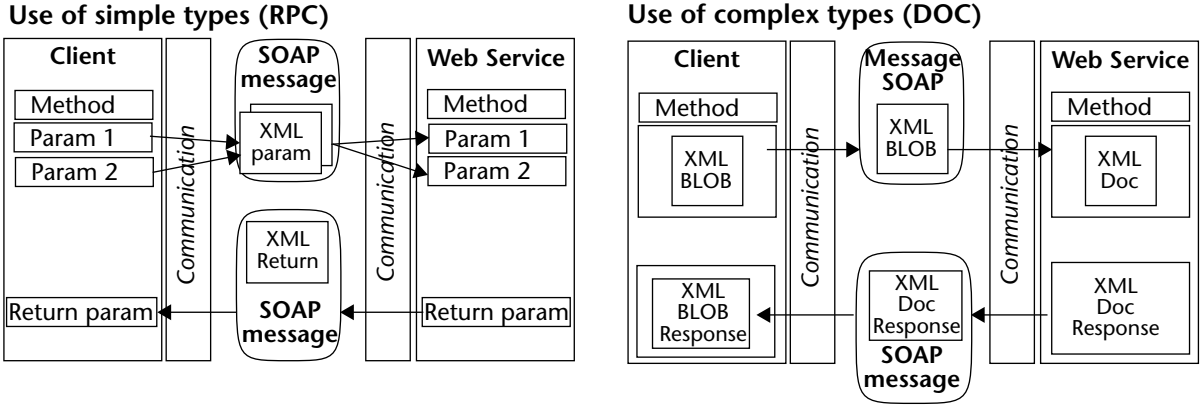
Processing complex types

4th Dimension enables you to use Web Services published in either RPC or DOC mode and including complex types (see the paragraph "Compatibility of RPC, DOC and complex types," page 727).

Note Despite the fact that they are complex XML types, data arrays are handled by 4D as simple types.

The proxy methods generated by the Web Services Wizard that include complex types (i.e., published in RPC mode with complex types or in DOC mode) are similar to standard proxy methods. However, you will notice that with these Web Services, the CALL WEB SERVICE command includes, as a parameter, a constant containing the word *manual*.

In fact, the use of such Web Services requires additional processing by the 4D developer. The main reason for this is that complex types are exchanged in the form of documents or XML *elements*. This means that in order to extract or include information in these SOAP parameters, prior XML parsing is necessary — whereas in the case of simple types, the parameter values are directly readable.



In 4th Dimension, complex type parameters (except for arrays) are handled in the form of BLOBs. You must use 4th Dimension language commands to manage them.

Complex input and output types

Web Services published in RPC mode can offer every possible combination between simple/complex types and input/output parameters. For instance, a Web Service may accept requests containing parameters of the simple type and send results in the form of complex types back to the client.
Conversely, Web Services published in DOC mode only handle parameters of the complex type.

The processing of complex types differs depending on whether the SOAP parameters are input and/or output parameters:

- For input parameters (received by 4th Dimension), the parsing of the XML structures and the extraction of useful information can be performed using the 4th Dimension XML commands (see the *Language Reference* manual).
- As far as output parameters (sent by 4th Dimension) are concerned, it is up to the 4th Dimension developer to "manually" build the XML structure to be sent to the Web Service, according to the information required by the server.

For more information about this operation (as well as the parsing of input parameters), we recommend that you refer to the code of the sample databases supplied with 4th Dimension. For more information about sending and/or receiving complex type parameters, refer to the description of the CALL WEB SERVICE command in the *4th Dimension Language Reference* manual.

A Segmenting Data Files

4th Dimension allows you to create data files as large as 128 gigabytes. However, no current microcomputer operating system supports this file size and hard disks of this capacity are not available. For these reasons 4th Dimension and 4D Server allow you to partition your data file into a maximum of 64 segments of 2 gigabytes each. Each segment can be located on a different physical volume. To increase the size of the data file beyond 2 gigabytes, you add data segments. This allows you to place different segments of a data file on different physical volumes.

Segmenting Data Files

You can either segment a data file at the time you create the database or after you begin to use it. You will want to segment a new data file if you expect the data file to become very large. Segmenting a data file allows a virtually unlimited amount of data to be stored.

Note You do not need to create any data segments unless you have more than 2 gigabytes of data or your hard disk cannot accommodate the size of your data file.

When segmenting a data file, you divide the data file into segments and then specify on which volume each segment is to be stored. For example, 4 gigabytes of data could be divided into 2 segments of 2 gigabytes each. Each segment can be limited in size, so you can reserve space on your hard disk for other files and avoid a completely full volume.

4th Dimension transparently fills the data segments in the order in which they were created. When a segment is full, 4th Dimension automatically moves to the next one. If by deleting records you make room in a data segment, the holes created in the segment will be reused.

When all of the data segments are full, you will be prompted with a message stating that there is no more room on the volumes where the segments are located. At this point, you would want to add a data segment.

Segmenting a Data File

You can segment a data file at any time.

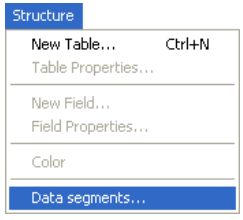
To increase the size of an existing data file beyond 2 gigabytes, you can add data segments, each of which can contain up to 2 gigabytes of data.

You create segments for existing data files in the Design environment. If you are using 4D Server, you create and manage data segments on the server machine.

► To segment a data file:

1 **Display the Structure editor window (in 4th Dimension stand-alone only).**

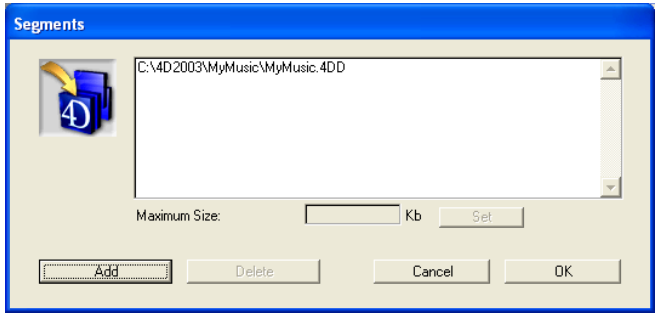
2 **In 4th Dimension, choose Data Segments... from the Structure menu..**



OR

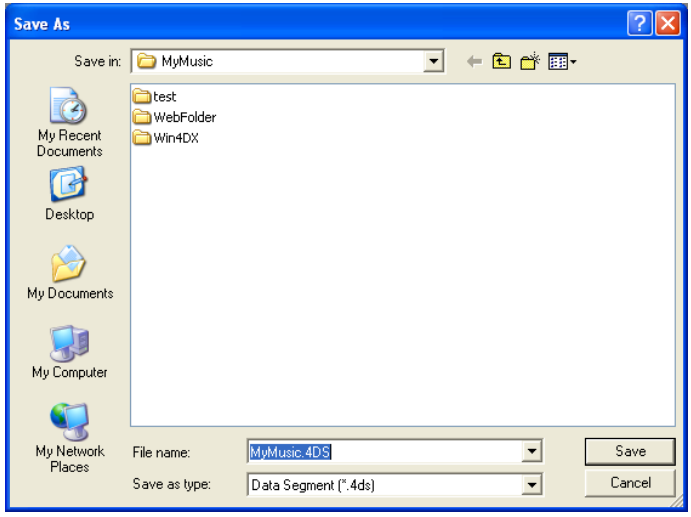
Select Segments... from the Data menu (4D Server).

The Data Segment Management dialog box appears.



3 **Click the Add button to create a segment.**

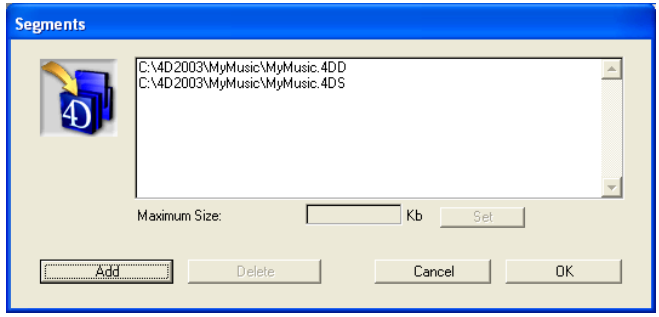
4 **Enter the segment name and select its location.**



4th Dimension provides a default name for the segment, which is the name of the database followed by the number of the data segment, followed by the prefix".4DS" (on Windows) or.data (on MacOS). This naming convention allows you to easily distinguish each data segment. You can change the names of the data segments at any time.

5 **Click the Save button.**

The Data Segment Management dialog box reappears, displaying the new data segment.



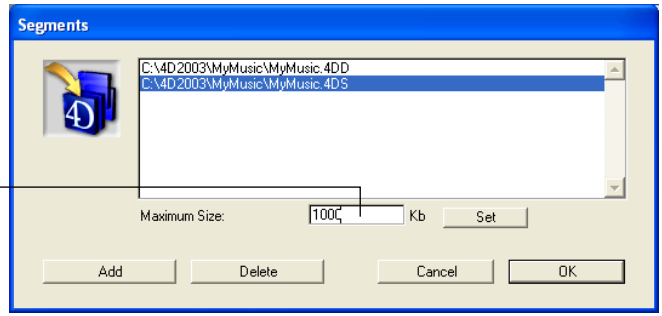
Limiting Data Segment Size

If you do not specify a size limit for data segments, 4th Dimension will fill the data segment until the volume on which it is located is full or until the limit of 2 gigabytes is reached.

► To specify a maximum size for a segment:

1 **Select the data segment in the Data Segment Management dialog box.**

2 **Enter a size (in kilobytes) in the Maximum Size box.**



Maximum Size box

3 **Click the Set button.**

This limits the size of the segment and allows you to reserve space on your drive for other files.

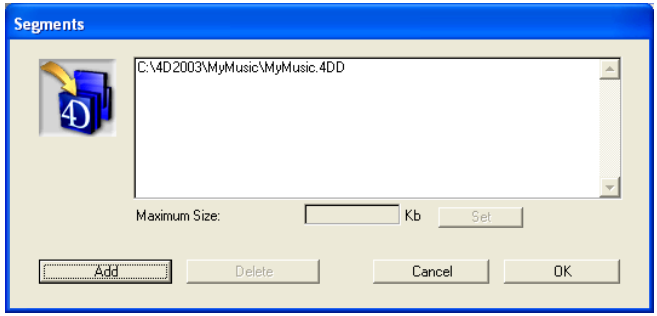
Note You can change the limit of a data segment that already contains data. In this case, the limit cannot be less than the size of the data already present in the segment. If you specify a lower size, 4th Dimension will automatically adjust the limit to the current size of the data file when you validate the new limit.

Adding a Data Segment While Indexing

If 4th Dimension reaches the maximum size of the last data segment during an indexing operation, it will present a dialog box allowing you to change the segmentation of the data file.

4D Server If you are using 4D Server, this error message appears on the server machine.

When you click the **Add Segment** button, 4th Dimension displays the Data Segment Management dialog box, which allows you to change the maximum size of a data segment or add new data segments.



Note The Add Data Segment button is available only if you have access privileges to the Design environment. For more information on access privileges, see Chapter 10, "Managing Password Access" on page 613.

Deleting Data Segments

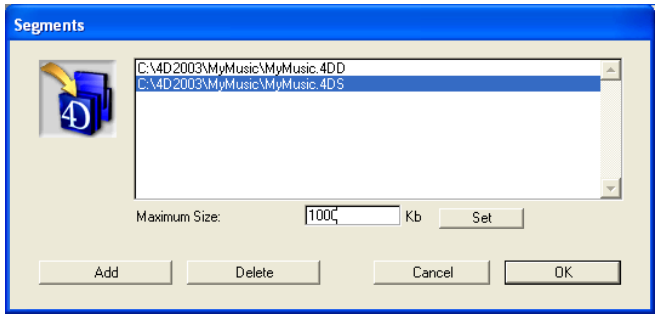
You may want to delete data segments if a large database suddenly decreases in size, or if you initially created more segments than are ultimately necessary. The method that you use to delete a data segment depends on whether it is a new or existing data segment.

Do not delete a data segment in the File Manager or the Finder. For more information, see the section "Missing Data Segments" on page 754.

Deleting a New Data Segment

A new data segment is a data segment that you just created in the Data Segment Management dialog box. You can delete a data segment only while the dialog box is still open. Once you leave the dialog box, the segment becomes part of the data file (in this case, see the following section).

To delete a data segment, click the segment you just created and then click the **Delete** button.

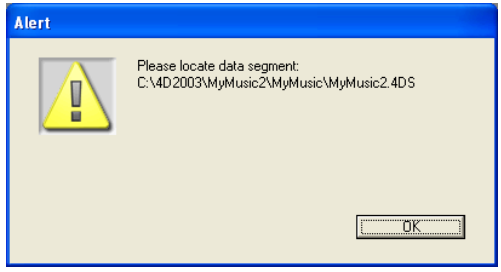


Deleting an Existing Data Segment

You can delete an existing data segment by using 4D Tools to compact the data file. Once you compact the data file, you can reconfigure the segmentation of your data file. For more information on reconfiguring data segments, see the section "Reconfiguring Data Segments" on page 755

Missing Data Segments

Do not delete data segments using your operating system! Do not use the File Manager or the Finder. If you delete a data segment using your operating system, 4th Dimension will ask you to locate the missing data segment.



Click the **OK** button. An Open File dialog box appears. If you no longer have the data segment, click the **Cancel** button. Since you could not locate the data segment, 4th Dimension will not allow you to open the database. In this case, you will need to use 4D Tools to repair the database. 4D Tools is a utility that is supplied with 4th Dimension applications.

Open your database with 4D Tools. The application will ask you to find the missing data segment. Click the **OK** button. In the Open File dialog box, click the **Skip** button.

4D Tools displays an alert stating that it is going to repair the database by tags. Click the **OK** button. Another dialog box appears in which you click the **OK** button to repair the database by tags.

When you repair by tags, 4th Dimension recreates the data file with the segments present on your disks. This new data file will allow you to reopen your database, but does not guarantee the integrity of your data.

Reconfiguring Data Segments

You may want to reconfigure the segmentation of your data file to accommodate a change in disk size or to change the number of data segments.

For example, suppose you have a 1 gigabyte data segment and you need to switch to two 500 megabyte drives. To do this, you must divide the 1 gigabyte segment into two segments.

The most efficient way to reconfigure data segments is to compact your database using 4D Tools. This method is described in this section.

Before you begin, be sure that you have enough room on your disk for another copy of your database. The process of compacting a database creates a new copy of both the structure and data files. Also, because the compacting process can take some time, be sure to plan accordingly. We recommend that you give 4D Tools the largest possible memory allocation to quicken the compacting process.

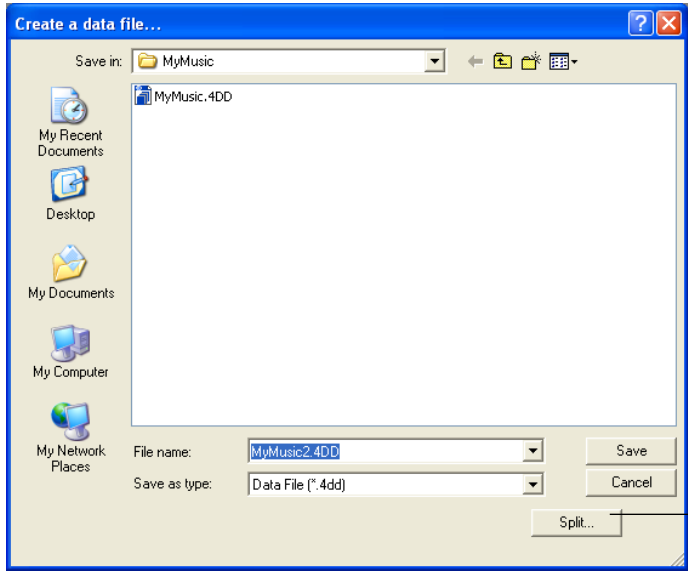
► To compact your database and reconfigure your segmentation:

1 **Launch 4D Tools and open your database.**

2 **Choose Maintain from the Tools menu.**
OR
Click on the Maintain tab.

3 **Click the Compact... button.**

A save-file dialog box containing a **Split** button appears.



4 Click the Split... button.

The Data Segment Management dialog box appears.

You can use this dialog box to add new data segments or to reconfigure the data segments as you wish.

Note For more information about 4D Tools, refer to the documentation for this utility.

B Assigning a Help File to a 4D Database

4th Dimension allows you to associate a custom on-line help file with each database.

The on-line help system provided with 4th Dimension is compatible with each 4D work environment (stand-alone database or client/server, interpreted or compiled, run using 4D Runtime, or integrated into a 4D Engine).

4th Dimension supports two help files formats which correspond to the industry standards (HTML and HLP).

Moreover, you can associate a precise section of this help file with each of the database's forms, which allows you to provide contextual on-line help.

Setting a Database On-line Help

File Formats

The format of the files must be one of the following:

- a ".HLP" file, which is the standard Microsoft Help[1] format for Windows. This type of file can be used on MacOS[2], provided the Microsoft Help for Mac application has been installed.
- a file ".HTM", which is the HTML format. With this format you can create an identical on-line help on all platforms. The on-line help is then displayed in a Web browser.

Note The Plug-ins can also have a help file, which must be placed in the Win4DX and/or Mac4DX folder for both single-user and client-server applications. The Plug-in's help file must be the name of the Plug-in with the .HLP or .HTM extension.

Assigning the Help File to the Database

Once the Help file generated, you need to associate it to a database so that it is opened when users call the on-line help. To assign a help file to a database, make sure

- The name of the help file is identical to the name of the database's structure file. It must also have the ".HLP" or ".HTM" extension, depending on its format (see above) and platform.
- The help file is placed next to the structure file or in the Win4DX and/or Mac4DX folder located at the same level as the database structure file.

4D Server If you want the help file to be accessible to all the client workstations, place the file in the Win4DX and/or Mac4DX folder. It will then be transferred to the client workstations automatically.

1. For information on how to generate a .HLP file, refer to Microsoft's documentation.
2. You have to change the document type to HELP and the creator to MSHE. Successfully converting the document depends on the version of the on-line help compiler as well as the version of Microsoft Help being used. Otherwise, this solution doesn't allow you to use contextual on-line help.

Creating Contextual On-line Help

Creating contextual on-line help is done by associating a section number with each of your forms. When you call the on-line help from a form, the corresponding help page is displayed. When the user calls the on-line help, the help topic that has the same ID as the form is displayed.

Contextual on-line help is available:

- under Windows, if the on-line help is in HLP format.
- under Windows and MacOS, if the on-line help is in HTML format.

The assignment of an ID number to a form is done in the form properties, for more information, refer to "Contextual On-line Help" on page 301.

Once you have assigned the ID number to the form, you need to assign the same ID number to the help file. This operation varies according to the format you are using:

- for HLP files, refer to the Microsoft on-line help compiler's documentation (Help Compiler).
- for HTML files, you must declare each section and assign a number. A section is declared by using a marker of type <A NAME="Section">. For example <A NAME="21">.
  The URL of the section has the following form <A HREF="#Section">...</A>. For example <A HREF="#21">...</A>

If the section number passed in the form is 0 or if it does not exist in the file, 4D displays the first page of the help file.

Calling the On-line Help from a Database

In 4D, you can call a database's custom on-line help in two ways:

- by choosing **DatabaseName Help** in the **Help** menu. In this case, the first page of the help file is displayed.
- by pressing the **F1** key (MacOS and Windows) or the **Help key** (MacOS only) when a form is displayed on screen. In this case, if a help field number has been associated with the form, the corresponding page is displayed (contextual help); otherwise the first page of the help file is displayed.

C DTD for macros

This appendix supplies the DTD (*Document Type Declaration*) for the macros file (Macros.xml) used by the Method editor of 4th Dimension. For more information about the use of macros, refer to the paragraph "Creating and using macros," page 577.

You can use this DTD for:

- precise display of the XML syntax and grammar for the markers used in the macros file.
- validation of a customized macro file in order to ensure its conformity with XML specifications.
  To do this, simply recopy the DTD in a text file that you will name, for example, "Macros.dtd". You must then use an XML parser (for example, the XML commands of 4th Dimension). For the parser to take the DTD into account, simply add the following statement to the first line of the macros file:

  <!DOCTYPE macros SYSTEM "c:\macros.dtd">

  (... where "c:\macros.dtd" specifies the pathname of the DTD file that you have created)

DTD for Macros:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Grammar for 4th Dimension methods macros. 4D SA.-->
<!-- caret position after macro expansion -->
<!ELEMENT caret EMPTY>
<!-- placeholder for selection -->
<!ELEMENT selection EMPTY>
<!-- placeholder for operating system user name -->
<!ELEMENT user_os EMPTY>
<!-- placeholder for 4D user name -->
<!ELEMENT user_4d EMPTY>
<!-- placeholder for method name -->
```

```
<!ELEMENT method_name EMPTY>
<!-- placeholder for current time -->
<!-- format = 0 to 8 -->
<!ELEMENT time EMPTY>
<!ATTLIST time
   format CDATA #IMPLIED
>
<!-- placeholder for current date -->
<!-- format = 0 to 6 -->
<!ELEMENT date EMPTY>
<!ATTLIST date
   format CDATA #IMPLIED
>
<!-- placeholder for clipboard contents -->
<!-- index = 0 to 9 -->
<!ELEMENT clipboard EMPTY>
<!ATTLIST clipboard
   index CDATA #IMPLIED
>
<!-- macro contents -->
<!ELEMENT text (#PCDATA | user_os | user_4d | time | date | method_name | caret |
selection)*>
<!-- macro -->
<!-- name = name as displayed in contextual menu -->
<!-- type_ahead_text  = text to type to activate the macro using type ahead (default
is the macro name) -->
<!-- type_ahead = should this macro be available using type ahead? (default is true)
-->
<!-- note: if the macro contents uses the selection placeholder, it cannot be activated
using type-ahead  -->
<!-- in_menu = should this macro be displayed in contextual menu? (default is true)
-->
<!ELEMENT macro (text?)>
<!ATTLIST macro
   name ID #REQUIRED
   type_ahead_text CDATA #IMPLIED
   type_ahead (true | false) "true"
   in_menu (true | false) "true"
>
<!-- a macro file contains macros -->
<!ELEMENT macros (#PCDATA | macro)*>
```

Index

Symbols

! (exclamation point), in entry filter codes 377
" (quotation marks), in entry filters 382
(number sign)
  as a placeholder 400
  in display formats 400
  in entry filter codes 377
  in entry filters 383
  placeholder in Alpha fields 405
$ (dollar sign), in display formats 401
$4D Compiler 674
$Runtime explorer 674
%R-, %R+ (special comments) 707
& (ampersand character)
  displaying a number as a time 402
  for scientific notation 402
  in entry filter codes 377
  initiating an entry filter 381
() (parentheses), as dead characters 384
* (asterisk) as a placeholder 400
.4dc 716
.c4d (method export) 586
.comp 716
.rsr 716
.sym (Windows symbol file extension) 138, 701
.symb (MacOS symbol file extension) 138, 701
.xml (error file extension) 138, 703
:= (assignment operator) in methods 529, 533
; (semicolon)
  in number fields 401
  specifying display formats for embedded fields and variables 495
< (less than character)
  in embedded fields and variables 494
  over maximum number of digits 400
<!--4DVAR MYVAR--> (Web server) 148
<?[TableName]> (Dynamic Table name) . . . 337, 370
<?[TableName]FieldName> (Dynamic Field name) 337, 370
> (greater than character) in embedded fields and variables 494
@ (Management) 135
@ (wildcard character)
  in entry filters 379
  in Method editor 567
  wildcard searches 212–214
[] (brackets), for table names 529
[MYVAR] (Web server) 148
\ (escape character) 575
^ (caret)
  as a placeholder 400
  generating a non-breaking space 400
_ (underline character), for placeholders 383
` (comments) 540, 571
| (vertical bar character)
  custom display format 403
  for display formats and filters 385
  for entry filters 385
~ (tilde character)
  forcing capitalization 381
  in entry filter codes 377
  initiating an entry filter 381

Numerics

1919382119 (random value) 139
3D buttons (Variable type) 139
3D check boxes 426
  Variable type 139
3D radio buttons (Variable type) 139
4D Backup 37
4D Chart
  Commands 555
  Plug-in areas 451
4D Client 28, 228
  access to Web server) 636
  opening a password-protected database from . 616
  using Process List editor from 680
4D Connect (connection to Web server) 149
4D Customizer Plus 117
  linking structure and data files 36
  Size of splash screen 611
4D Engine
  Customizing 720
  Folder 718
4D Folder 81, 451, 452

4D Help 81, 550
4D Internet Commands 555
4D Modèles 29
4D OLE Tools. 281
4D Open 159
4D Open Access (Preferences) 120
4D Open connections (allow) 120
4D Runtime 612, 712
4D Server
  4D Client Timeout 119
  4D Open Access. 120
  changing form names 265
  changing method names 543
  creating data segments with 750
  Encrypting connections 119
  field properties 176
  modifying database objects. 105
  modifying menus 591
  modifying table definitions 170
  moving table images 163
  object locking while modifying forms . . .268, 289
  object locking while setting preferences. 107
  object locking, with lists 642
  object locking, with methods 534
  object locking, with passwords 614
  Preferences. 118
  Process List 680
  resizing a table image 162
  saving form pages 362
  saving server path 616
  setting maximum and minimum values 389
4D Templates. 29
4D Tools
  compacting a database with 755
  repairing a database with 754
4D View 452
4D WebSTAR (connection to Web server) 149
4D Write (text storage) 178
4DACTION 544
4DMETHOD 544
4DWSDL 731
4th Dimension
  API 120
  application icon 29
  confusing. 289
  flushing data buffers 116
6.8 compatibility for text rendering (Preferences .111

A

A_WebService 151
Abort icon 686
About locked files (Preferences). 136
Accept button 424
Access drop-down list
  function of. 543
  in Form Properties window 291, 634
  in Method Properties window. 635
access group
  setting for forms 634
  setting for menu commands 635
  setting for methods 635
Access Management (Preferences) 132
access privileges
  creating an access hierarchy 617–619
  for 4th Dimension environments 619
  for Design environment 621
  for forms 291–292, 633
  for menu items 635–636
  for methods 543, 634
  for plug-ins 636–637
  for record operations 167, 631–633
Access to Plug-Ins dialog box 637
Accessing a Web Service 734
ACCUMULATE command 501
Action drop-down menu (for creating buttons) . 423
Activate Automatic Comments 88
  Preferences 131
Active 4D folder 81, 451, 452
active objects 419–455
  automatic numbering of in grids 460
  changing appearance of 334–345
  copying 328–329
  creating 414–419
  creating arrays of 459–460
  definition of. 366
  deleting 331–332
  display formats for. 415–416
  duplicating. 326
  enterable 420–421
  groups of 315
  keyboard shortcuts for. 609
  non-enterable 420–421
  on forms 412
  role of. 230–231

setting data entry controls 417
setting display formats 415
types of 366, 419–460
active windows (working with multiple) 673
Add Field tool/icon 275, 367
Add Item (Menu menu) 597, 606
Add Menu menu item 602
Add record operations (access to) 632
Add to Subform button 203, 425, 471
adding
fields to form (from Explorer) 68
form name to method (from Explorer) 71
subform to a form (from Explorer) 71
table names and field names to methods (from Explorer) 68
addresses 692
Administrator
access to Password editor 638
as group owner 620
system maintenance role 637
using Edit User dialog box 625
Advanced button (in Object Properties window) 451
Advanced options (in Form Wizard) 233
Advanced parameters 741
Advanced screen (in Form Wizard) 243–262
ALERT command 528
Align menu item 280
Align to Grid menu item 280, 321, 322
aligning objects 316
using alignment grid 320
using alignment tools 317–318
alignment
setting for text objects 340
tools in Tools palette 317
using Alignment Assistant 318, 323
using alignment grid 320
alignment tools
Align Bottom tool 317
Align Center Horizontal tool 317
Align Center Vertical tool 317
Align Left tool 317
Align Right tool 317
Align Top tool 317
All variables are typed (Preferences) 140
Allow 4D Open Connections (Preferences) 120
Allow 4D WebSTAR to connect via 4D Connect (Preferences) 149
Allow Deletion Control (Preferences) 135, 209
Allow Drag (Preferences) 123
Allow Expand/Collapse (Preferences) 123
Allow Read Only Data file Use (Preferences) 136
Allow SSL for Web Server (Preferences) 142
Allow Type-Ahead (Preferences) 123, 567
Allow Web Services Requests (Preferences) . 151, 729
Alpha field formats 404
Alpha fields 177
Alt+Click (open object method) 465
Always compile for both platforms (Preferences) 137
Always Yes for All (Forms) 128
ampersand (&) character
displaying a number as a time 402
for scientific notation 402
in entry filter codes 377
initiating an entry filter 381
Appearance drop-down menu (in Form Wizard) . 258
Appearance settings
for buttons 422
for check boxes 427
for text objects 339
in Form Wizard 258
Append Menu menu item (Menu menu) 596
Application file 719
Application Main Memory (Preferences) 117
Application Preferences 114
applications
building 51
building (double-clickable) 717
customizing icon 720
defining name 715
protection of 692
arguments
format of 530
in entry filters 381–383
in methods 530, 533
arrays
compiler methods (Preferences) 140
in symbol table 702
arrow keys (moving objects) with 305
Arrow tool/icon (selecting objects) 302
Ask (Forms) 128
assignment operator (:=), in methods 529, 532
Associating a Comment to an Object 85
asterisk (*) as a placeholder 400
Auto Assign Related Value in Subform 203–204
in Relation Properties window 203
Auto Delete Related Many 206
Auto One to Many check box 203
Auto Platform (Interface) 110

Auto Relate One check box 201, 203
Auto Wildcard Support check box 202
Automatic
  Form Wizard 249
  Platform interface 110
automatic actions 424
automatic buttons
  buttons 421
  Highlight buttons 422
  Invisible buttons 423
Automatic Comments 88
automatic foreground and background colors . . .344
Automatic Form Creation (Preferences) 128
automatic relations 208, 210, 468
  Auto Assign Related Value 203
  establishing 208
  Many to Many 215
  One to Many 196
  One to One 214
  records selected in related table 220
  See Also relations
  turning off 203
automatic repositioning 309
automatic resizing 309
Automatic Transactions during Data Entry
    (Preferences) 135
Automatic Width checkbox (for subforms) 475

B

background color 346
background graphics in picture fields 410–411
background page
  displaying 352
  in Detail forms 230
  on multi-page forms 350
  selecting objects on 302
backing up, databases 36–37
Backslash (escape character) 575
backup systems (for databases) 36
Balloon Help 188
Basic screen (Form Wizard) 232–233
Batch setting of attributes 544, 545
  project methods 75
Best Match (Platform interface) 110
Blob fields 180
Block Size (Preferences) 117
Boolean field formats 405
Boolean fields 179
  default value 407
  formats 405
  formatting as check boxes 407
  formatting as radio buttons 406
Border menu item 279
borders
  setting for form pages 362
  setting patterns of 343
brace matching (Method editor) 561–562
brackets ([]), for table names 529
Break areas 480, 482, 506, 507
  additional 480, 482
  additional levels of Breaks 507
  calculating subtotals 501–503
  calculating totals in reports 491
  creating additional 482
  in output form 477
  in printed reports 482, 491
  in reports 491, 507, 508
Break control lines 482, 492
  creating additional 499–501
  deleting 501
Break Header areas 499, 506
Break Header control lines 500
  adding 499–501
  deleting 501
BREAK LEVEL command 501
break levels 499
  additional 503–504
Break List (modification) 93
Break Page (Runtime Explorer) 93
break processing 500
  initiating 501
breaks
  examples of calculations in 504
  for calculations 502–503, 505
  multiple 499
Bring to Front icon 688
Build (button) 719
Build application 713
  File menu 55
business rules,enforcing 371
button grid 439–440
  Variable type 139
Button Text area 421

buttons 421
3D 422
as active objects 421–425
creating 423
default type (Compilation) 139
for subform actions 425
for subforms 471
Highlight 422
Invisible 423
labeling 421
pulsing 102
pulsing (MacOS X) 422
radio 427–429
standard 421
variables associated with 423
Buttons page (Form Wizard) 245, 354

C

Cache
maximum 116
minimum 116
statistics (Runtime Explorer) 92
Web Server 145
Cache Manager process 673
calculations
in reports 491
methods for 420–421, 462–467
object methods for 508
subtotals in reports 501–503
CALL PROCESS command 525
Can't Delete if Related Many option 205
Can't Modify attribute 183
assigned to primary key field 206
in Field Properties window 372
in related tables 196
Cancel button (automatic action) 424
capitalization (tilde (~) character for) 381
caret (^)
as a non-breaking space 400
as a placeholder 400
Case Sensitive 583, 584
Find Editor option 97
case sensitivity, in passwords 625
Catch Page (Runtime Explorer) 93
characters
dead 384
in entry filters 383
in filter arguments 382
sending extended ASCII 147
Standard Set (Web) 147
to initiate a filter 381
using control characters (in menu labels) 597
Check Boxes 426
check boxes 426–427
default value of 426
formatting Boolean fields as 179
values in 426
variable type 139
Check syntax option (Compiler) 695
Choice List drop-down menu 376
choice lists 372, 374
assigning at form level 374
assigning in Field Properties window 372
assigning to a field 175
See also lists
uses of 641
Choices & Help page (Field Properties window) . 188
Choices attribute (Field Properties) 188
circular relations 221–222
Clear (compiled code) 698
Clear (standard action) 425
Clear Cache (Web Server) 146
Clear menu item 331
in Edit menu 57
Clear Object Method menu item 281, 467, 541
Client/Server
Communication 119
Connections Timeout 119
Preferences 118
Timeout 119
Clipboard History 570
clipboards
copying objects to 57
numbering of 570
pasting pictures into Picture fields 409
storing copied objects 329
storing deleted objects 331
viewing contents of 57
Close *EditorName* menu item 61
Close *FormName* menu item 362
Close Structure menu item 56
closing all Design environment windows 61
codes (entry filter) 380
Collapse all 568

color
in forms . . . 344
in methods . . . 558
setting for a field . . . 189
Color Background (Preferences) . . . 125
Color menu item . . . 279
Object menu . . . 344
Color Names (Preferences) . . . 125
Color page . . . 189
Field Properties window . . . 175
Relation Properties window . . . 209
Color property (Relation Properties window) . . . 206
Colors page . . . 343
Object Properties window . . . 341, 344, 347
combo box
managing a . . . 438
setting default values for . . . 390
Command key (reserved key combinations) . . . 607
Command name (function) . . . 80
commands
adding to a method . . . 81
DISABLE ITEM . . . 603
displayed in Method editor . . . 530
displaying on-line documentation . . . 77
displaying the documentation . . . 79
New process . . . 600, 675, 682
role of . . . 528
Commands page (Explorer) . . . 77–81
Comment/Uncomment . . . 571
comments . . . 571
%R- and %R+ . . . 707
automatic comments . . . 88
for published methods . . . 734
inserting date, time, or user . . . 87
Preferences . . . 131
compacting databases . . . 755
compatibility
of structure and data files . . . 34
Preferences . . . 128
Use V3.x.x File Procedure Scheme . . . 129
Use V3.x.x Startup Method Scheme . . . 128
Web (Preferences) . . . 148
Compilation
All variables are typed . . . 140
Always compile for both platforms . . . 137
Benefits of (execution speed) . . . 691
Checking code . . . 692
Compiled objects . . . 690
Compiler Methods . . . 140
Default Alpha Type . . . 139
Default Button Type . . . 139
Default Numeric Type . . . 139
Fat binary mode . . . 691
Generate the Symbol File . . . 138
Generating error file . . . 138
Initializing Local Variables . . . 139
Preferences . . . 137, 700
Process and interprocess are typed . . . 140
Range Checking . . . 138
Type the variables . . . 139
Compilation Path (Preferences) . . . 139
Compile button (Compiler) . . . 695
Compiled
Code (Clearing) . . . 698
Database (building) . . . 716
Difference between interpreted and compiled 689
Objects . . . 690
Compiler . . . 50
Definition . . . 689
Options . . . 698
Compiler Methods (Preferences) . . . 140
Completely Delete option . . . 170
Complex types . . . 727
Processing of . . . 745
Components . . . 83
Components Page (Explorer) . . . 83
Connect Menu menu item (Menu menu) . . . 602
connected menus . . . 595
advantages of . . . 602
creating . . . 602
deleting . . . 603
modifying . . . 602
Connection parameters (Web Services) . . . 744
Consider @ as a character for Query and Order By . . . 135
constant (adding to method) . . . 76
Constants page (Explorer) . . . 76, 465
Contents of SOAP Action field . . . 735
Contextual menus (Structure window) . . . 60
Contextual On-line Help . . . 301
control lines
creating additional . . . 499–501
in forms . . . 480–484
in reports with several breaks . . . 507
moving output . . . 483
Control page (Relation Properties window) . . . 208
Coordinates page (Object Properties window) . . . 306, 310

Copy (standard action) 425
Copy menu item 329
in Edit menu 57, 607
copying form objects 328–329
CPU Priority (Preferences) 115
Create table (data import) 132
creating
active objects 414–458
buttons 423
check boxes 426
custom formats and filters 385–388
custom labels 493
database structures 159–160
databases 24
enterable and non-enterable objects 420–421
fields 172–176
forms 235–240
labels 493
lists 82, 646–647
menu bars 593
methods 534
object methods 534
radio buttons 427–429
relations using Explorer 68
table relations 208–209
crosshair/crossbar pointer for creating objects . . 313
current date (as default value) 390
Current Item area (List editor) 651
Current Item Properties (Menu Bar editor) 599
Current List area 647
in List editor 648–650
Current Menu Bar list 606
for deleting a menu 609
in Menu Bar editor 594
Current Menu Item area 595, 604, 677
current record 220
current selection
in unrelated tables 256
working with multiple 673
current time (as default value) 390
custom applications
definition of 40
in Custom Menus environment 40
menu bars for 601, 611
custom entry filters 385
custom formats (for Alpha fields) 405
custom formats and filters
creating 385–388
editor 397
custom menus
adding menu items 596–597
adding splash screens to 610–611
adding to menu bars 595
changing font styles 603–605
custom applications and 611
example of 589
keyboard shortcuts for 607–609
previewing 610–611
standard action 426
Custom Menus environment
applications for 40
definition of 40
with menus 612
Custom Menus item (Use menu) 40
custom toolbar 591
Customizing
the application icon 721
the application signature 720
Cut (standard action) 425
Cut menu item 57

D

dashes (as dead characters) 384
Data
Allowing Deletion Control 135
Automatic saving 116
Data Control (Preferences) 134
Data Control page (Object Properties window) 388, 389, 397
data entry
Automatic Transactions 135
default order 358
options for subforms 470
data entry controls
choice lists 374
default values 389
entry filters 376
excluded lists 376
for enterable objects 417
Mandatory attribute 374
maximum and minimum values 645–646
setting 371
using lists as 645
data entry order 358–361
entry order pointer 359
grouping objects 361
restoring standard order 361

setting first object 360
viewing and modifying 359–360
data files
Allow Read Only use (Preferences) 136
creating new 35
locating 32–35
maximum size of 749
naming 33
paths to 35
segmentation (reasons for) 749
segmenting existing 750–751
selecting 32
data segment
adding while indexing 752–753
deleting a new 753
deleting an existing 754
Data Segment Management dialog box 750–751, 752, 753
for reconfiguring data segments 756
data segments
adding 749
dealing with missing 754–755
deleting 753, 754
limiting size 751
reconfiguring 755
Data Segments menu item (Structure menu) 750
Database methods
On Exit 524
On Server Close Connection 524
On Server Open Connection 524
On Server Shutdown 524
On Server Startup 524
On Startup 524
On Web Authentication 524
On Web Connection 524
database methods
in symbol table 703
modifying 73
role of 518, 521
writing 524
database structure 153
creating 159–160
fields 154
flat file 156
multiple table 156
structure files 32
subtables in 180–181
table relations 157
Database theme (Preferences) 132
databases
backing up 36–37
building applications 55
compacting 755
creating 24
creating structure of 159
definition of 154
fields in 154
flat file 156
multi-process 673
naming 28
opening 29
publishing at startup 142
records 154
relational, defined 157
relations 38
selecting a different data file 32
structure of 153
tables in 155–159
using templates 26
Date fields 179
display formats for 398
dead characters (in entry filters) 384
Debug window 686
debugging processes 686
decimal points, in number display formats 400
Default Alpha Type (Preferences) 139
Default Button Type (Preferences) 139
Default Display (Method editor) 122
Default Font (Preferences) 121
default fonts 281
default forms (automatically created by 4th Dimension) 228
Default Home Page (Preferences) 145
Default HTML Path (Preferences) 144
Default HTML Root (Preferences) 144
default lists of values 390
default menu bars 592
default names, with processes 682
Default Numeric Type (Preferences) 139
Default Owner of Objects Created by This User (drop-down list) 626
default processes 673
default settings (for text objects) 338
default startup environment 127
Default User (Passwords) 622
Default User (Preferences) 133
Default Value entry area (Object Properties window) 389

default values
automatically generating . . . 390
Boolean fields . . . 407
setting . . . 389–391
setting programmatically . . . 392
using current date . . . 390
using current time . . . 390
using sequence numbers . . . 390
Default Values dialog box (Object Properties window) . . . 391
Default Window Size area (Form Properties window) . . . 298
default window size options . . . 297
Define Grid menu item . . . 320
Define Ruler Units menu item . . . 312, 513
Delete button
for deleting forms . . . 264
keystroke equivalent for . . . 331
Delete Item menu item (for deleting menu items) . . . 610
Delete menu item (for lists and list items) . . . 650
Delete Menu menu item (Menu menu) . . . 609
Delete Page menu item . . . 353
Delete Record buttons . . . 424
delete record operations (access to) . . . 632
Delete Related Many option (Relation Properties window) . . . 205
DELETE SELECTION command . . . 170
Delete Subform button . . . 425, 471
deleting
connected menus . . . 603
fields . . . 173
form pages . . . 353
forms . . . 264
list items . . . 650
lists . . . 650
menu items . . . 610
menus . . . 609–610
methods . . . 540
object methods . . . 467, 541
objects . . . 331
related records . . . 205
deletion control
in Relation Properties window . . . 205
in several related tables . . . 206
options . . . 209
Demo application . . . 718
Design (standard action) . . . 426
Design environment . . . 37–38
access to . . . 621
editors in . . . 41–49
menus . . . 54–59
object locking in . . . 105
role of . . . 38
toolbar . . . 62–63
Design item (Use menu) . . . 40
Design menu . . . 58
Edit Form menu item . . . 58, 69
Edit Method menu item . . . 58, 536
list of open windows . . . 58
New Form menu item . . . 58, 69, 235, 484
New Method menu item . . . 58, 535
Design Mode (Preferences) . . . 120
Design process . . . 673
Designer
access to environments . . . 619
assigning a password to . . . 623
role of . . . 619
system maintenance role . . . 637
users and groups created by . . . 620
using Edit User dialog box . . . 625
Desktop files . . . 32
Destination Folder . . . 715
Detail area . . . 481, 506
explanation of . . . 507
in output form . . . 477, 479
in reports . . . 507
in screen display . . . 492
Detail control line . . . 492
Detail Form for Printing (form type) . . . 492
Detail Form for Printing (option) . . . 493
detail forms
definition . . . 225
for subrecords . . . 469
dials . . . 453–455
example of . . . 453
DISABLE ITEM command . . . 603
Disable Type-Ahead . . . 567
disabling password access system . . . 623
Discover button (Web Services Wizard) . . . 738
disk capacity (change of) . . . 755
Display
Comments in the Explorer . . . 85
Field type icons . . . 125
Grid . . . 271
Limits . . . 272
Marker Labels . . . 272

Markers 272
Object Properties 271
On-line documentation 77
Page 0 271
Paper (borders in Form editor) 271
Property List 271
Rulers 271
Runtime Explorer 90
Tools Palette 271
display characters
for Number field formats 400–404
in entry filters 383
display formats 396–411
Alpha field formats 405
Boolean field formats 405–408
Date field formats 397
for objects 415
in mail-merge templates 495
Number field formats 399–404
Picture field formats 409–411
Preferences 113
Time field formats 398
with entry filters 379
Display Only attribute 183, 373
in Field Properties window 372
Display page (Object Properties window) . . .336, 427
display pages
adding 350
selecting objects on 302
Display User List in Password dialog box (Preferences) 134, 616
Distribute tools (Tools palette) 322
DOC 727
Documentation
Displaying in the Explorer (MacOS) 79
Download missing HTML pages if necessary (Preferences) 130
dollar ($) sign (in display formats) 401
Double Clickable check box (for subforms) 474
Double-clickable application
Building 717
Rebuilding 719
Down One Level 329
Download missing HTML pages if necessary (Preferences) 130
drag and drop
drop area 112
for form objects 419
from Tables page of Explorer 68
in Explorer 67
drop-down lists (initializing) 390, 435
DTD for macros 761
duplicate fields 174
Duplicate menu item 280
Object menu 326
Duplicate on Matrix menu item 280, 460
Dynamic Table and Field Names 370

E

Edit
Former mechanism 595
Preferences 57
EDIT ACCESS command 638
Edit Form menu item 264
Design menu 69, 288
Edit Group dialog box 627
Edit Group menu item (Passwords menu) 627
Edit menu 56–57
Clear menu item 57, 331
Copy menu item 57
Cut menu item 57
Menu bar editor 595
Paste menu item 57
Select All menu item 57
Show Clipboard menu item 57
Undo menu item 56
Edit Strings button (Object Properties window) . 435
Edit User dialog box 638
adding users 624
changing user name or password 626
viewing database usage 638
Edit User menu item (Passwords menu) 626
editor windows 41–61
closing 56
Form editor 43–44, 269
List editor 47–48
Menu Bar editor 45–46
Method editor 44–45
Password Access editor 46, 624
Process List editor 680–684
reverting to last saved version of 56
saving contents of 56
Structure editor 41–42, 160
Enabled/Editable (list option) 655
Encrypt Client/Server Connections (Preferences) 119
Endpoint URL 742

Enter in List mode 643
Enter key. 564
Enter Password dialog box 615
Enterable attribute 373
Enterable check box
for subforms 474
in Object Properties window 373
enterable objects 420–421
adding scroll bars to. 392
data entry controls for 417
display formats for 415
in data entry order 360
Enterable Related Fields check box (in Form Wizard) 373
Entry control
Keyboard layout 392
Web 148
Entry Filter display area 377, 385
Entry Filter drop-down menu 377–379
entry filters 376–385
arguments 382–383
choosing. 377–379
codes. 377, 380–383
custom 385
dead characters. 384
display characters. 383
initiating 380
modifying. 379
placeholders 383
with display formats 379
Entry Order menu item 277, 359–361
entry order pointer. 359
environments
changing between 38
selecting startup 127
error file 703
errors linked to a specific line 705
general errors 705
generating 138
structure of 704
warnings. 705
error messages (password access system) 617
errors
during execution 565
syntax 564
Escape sequences 575
events 463–465
role of in methods 522–523
Events page
Form Properties window 300, 523
Object Properties window 437, 463, 523
using the. 463
Events tab (Object Properties window) 304
exclamation point (!) (in entry filter codes). 377
Exclude Forms (Find Editor option). 97
Exclude Methods (Find Editor option). 97
Excluded check box 376
excluded lists 376, 646
Execute Method dialog box 675
for starting a new process. 678, 682
Execute Method menu item. 678
in User environment 522
Execute On Client function 119
executing
methods 530–533
object methods. 496
execution speed 691
Exit Design Environment menu item (File menu). 61
Exit Design Environment when going to Custom Mode (Preferences) 127
Expand all. 568
Expand/Collapse (methods) 568
Explorer 63–82
adding a command to a method 81
adding a constant to a method 76
adding a field with 369
adding fields to forms 68
adding form name to method 71
adding method name to a method 75
adding subform to form. 71
adding table names and field names to methods 68
Batch setting of attributes 75
Commands page. 77–81
Constants page. 76, 465
creating a list 82
creating a new method. 72
creating new form 69
creating relations using 68
Delete, New, and Edit buttons 66
deleting a form 70
designating current input or output form 71
displaying comments 85
displaying on-line documentation 77
displaying the. 63
Displaying the documentation (MacOS) 79
editing a form. 70

Forms page 69–71, 264, 288
Input Form check box. 263
Lists page 82
Methods page 72–75
modifying a list 82
naming a form using. 289
Output Form check box 263
pages of 63
preview area (showing and hiding) 65
preview area icon 68
previewing a form 70
previewing a table image 68
renaming a method 74
renaming forms. 264–265
resizing the 66
setting current input and output forms . .262–263
Tables page 67
using drag and drop 67
using to add a constant to a method 465
using to name a form 290
using to rename objects 65
using to write methods 466
viewing a table image in Structure editor. 68
Explorer menu item (Tools menu) 63
Export method 55, 585
Extended characters (Web) 147

F

Fast Mode (new Index mode) 193
Fat binary. 691
field attributes 182–188
Can't Modify 183
Choices 188
Display Only 183
Enterable 373–374
Indexed 183–184
Invisible. 185
Mandatory. 182, 373–374
Text with Scroll Bar. 392
Unique. 184–185
field display formats
See display formats
field entry filters. *See* entry filters
field length (setting maximum) 174
field names 154, 174
changing 191
duplicate 174
dynamic. 370
maximum number of characters. 174
reserved words in. 174
truncated 174
valid 174
Field number (Runtime Explorer) 93
field properties
changing 191
modifying 190
saving. 175
Field Properties menu item 190
Field Properties window 173, 189, 372
assigning a choice list 374
Attributes page. 174
Can't Modify attribute 372
Choices & Help page 188
Color page 189, 190
Display Only attribute. 372
entering help text 188
for establishing data entry controls 372
Indexed attribute. 372
Invisible check box 185
Mandatory attribute 372
modifying fields. 190
setting data entry controls 371
setting field attributes 182
setting field types 176–181
Unique attribute 372
field types. 176–181
Alpha 177
Blob 180
Boolean 179
changing 190–191
Date 179
Integer 178
Long Integer 178
Picture 179
Real 178
role of. 172
Runtime Explorer 93
Text 177–178
Time. 179
fields. 154
adding to forms 354
adding with the Explorer 369
as active objects 230
assigning a color to 175
data entry controls for. 371–396
deleting 173
display formats for. 396–411

duplicate . . . 174
embedding in text areas . . . 493
foreign key . . . 195
grouping in Form Wizard . . . 241
help for . . . 174, 393
icons . . . 125
indexed . . . 183–184
inserting in forms, shortcuts . . . 495
invisible . . . 185
Invisible attribute . . . 173
maximum number of . . . 158, 172
modifying in forms . . . 369
naming . . . 174
object locking . . . 176
placing on forms . . . 367–368
primary key . . . 195
related . . . 195
removing from form in Form Wizard . . . 242
reordering in a group . . . 242
reordering in Form Wizard . . . 240
selecting for forms . . . 236, 237–240
selecting for output forms . . . 486
standardizing entries for . . . 188
types of . . . 176–181
ungrouping . . . 369
using object methods with . . . 462–467
Fields page (Form Wizard) . . . 244, 246
File menu . . . 24, 55–56
Close *EditorName* menu item . . . 56, 61
Close Structure menu item . . . 56
Open Database menu item . . . 55
Page Setup menu item . . . 55
Print menu item . . . 55
Quit menu item . . . 55
Revert to Saved menu item . . . 56
Save *EditorName* menu item . . . 56, 105
File procedure (Preferences) . . . 129
files
desktop . . . 24, 32
exporting methods . . . 55
generated . . . 719
help formats . . . 758
importing methods . . . 55
locked . . . 136
segmenting . . . 749
selecting different (for database) . . . 35
temporary (storing) . . . 117
Fill menu item . . . 279, 342
fill patterns (setting) . . . 342
Filters (Preferences) . . . 113
Final application . . . 718
Find
dialog box . . . 94
Edit Menu . . . 57
Go to Line Number . . . 585
in Database . . . 582
in Database (Edit Menu) . . . 57
in methods . . . 582
Find and replace in methods . . . 582
Find Editor
Object types to search . . . 96
Scope of the Search . . . 96
Searching Options . . . 97
Find Next (Edit Menu) . . . 57
Find Previous (Edit Menu) . . . 57
Find Same . . . 583
Edit Menu . . . 57
First Page button . . . 424
First Record button . . . 424
Fixed string . . . 139
fixed-frame printing
Picture fields . . . 511
Text fields . . . 512
flags (picture button) . . . 432
flat file databases . . . 156
flow of control structures . . . 529
Focusable property . . . 417
font attributes
setting default . . . 336, 338
setting default for form . . . 338
setting in Form Wizard . . . 244
setting with style sheets . . . 259, 340
Font menu . . . 281
Font page . . . 340
Object Properties window . . . 260, 340
fonts/font size
default . . . 281
for objects . . . 281
General Font (Preferences) . . . 121
Message Font (Preferences) . . . 121
Method editor (Preferences) . . . 122
Preferences . . . 120
footer
control line . . . 492
marker . . . 483
Footer area . . . 507
explanation of . . . 482
in output form . . . 477, 479

in printed reports 491
in reports 507
For…End for structure 691
foreground and background color (Form Wizard)244
foreign key field. 200, 201, 209, 212
automatic assignment during data entry 204
definition of 195
in Form Wizard 238
Form editor 43–44, 226, 267–269
adding display pages 351
alignment tools 316–318
choosing a form type 293
compared to Form Wizard 234
creating custom labels. 493
customization options 234
display menu command 270
displaying border lines in 362
displaying forms in 289
duplicating multiple objects 327
elements of 268–281
Font menu. 281
Form menu 277–278
grouping objects in 315
menus 277
modifying form objects. 302
Object menu 279–281
opening forms in. 288–289
overview of 227
rulers in 269, 311–312
showing or hiding elements 270
Style menu 281
Tools palette 273–276
window 269
Form event constants 527
Form event function 464, 527
form events 463–465, 525–527
activating in Form Properties window . . .300–301
form letters (creating) 493
Form menu 277–278
Define Grid menu item. 320
Define Ruler Units menu item 312
Delete Page menu item 353
Entry Order menu item. 277, 359
Form method menu item 277
Turn Grid On/Off menu items 278
form methods 300–301, 519
creating 73, 538
for setting data entry controls 373
modifying 71
role of. 518, 519
writing 525–527
form objects
aligning 316–322
assigning a color to 344
changing appearance of 334–345
copying 328–329
creating 312
deleting 331–332
duplicating. 326
grouping 315–316
layering 329–331
moving. 305
resizing. 307–309
selecting. 302–305
splitters 456
ungrouping 316
form properties
events. 300–301
Form Type 294
overview of 282–284
resizing options 297, 299
setting 289–301
setting window margins 297
sizing options 297
Window Title. 295
Form Properties menu item
Form menu 277, 289, 633
Form scaling 277
Ruler definition 278
Form Properties window 289
Access drop-down list 291
assigning access privileges. 633
assigning owner privileges 633
Events page 300
naming a form using. 289
Owner drop-down list 291
Platform Interface drop-down list. 293
setting the Platform Interface with 292, 293
Form scaling
Fixed Ratio. 334
Macintosh to Windows platform 333
Rescale Pictures 334
Windows to Macintosh platform 333
Form Scaling dialog box 333
Form Size (Screen Sizes). 253
form template 236
creating 258
Form Template drop-down list (Form Wizard) . . 258

Form Wizard 69, 226, 302, 337, 492
adding a subform with 255, 469
Advanced screens 244–246
Automatic (option) 249
Basic screen 232–233, 235
Buttons page 354
creating mail-merge templates with 494
Mac OS 7 (option) 249
Mac OS 9 (option) 249
Mac Theme (option) 249
overview of 226
removing a field 242
renaming a group 242
reordering fields 240
role of 43
selecting fields in 237–240
Subforms page 237
Use Dynamic Field names 371
used to create multi-page forms 351
using 235–240
Windows 3.11, NT 3.51 (option) 249
Windows 95/98/2000, NT 4 (option) 249
Format drop-down menu (Object Properties window) 397
Formatting page (Preferences) 113
Former Edit menu mechanism 595
forms
active objects on 366, 412
adding fields to 354, 367–369
adding pages to 350
alignment grid 320
Always Yes for All 128
Ask 128
Automatic creation (Preferences) 128
Break areas in 482
control lines in 480–484
copying objects on 328–329
creating a label report form 515
creating new 69, 235–240, 257
custom labels 493
custom mailing labels 493
default, created by 4th Dimension 228
definition of 43
deleting 70, 264
deleting pages of 353
designating as input or output 71
Detail area in 481
editing 70
Footer area in 482
form type 293
group access to 633
group ownership of 633
Header area in 480
input 228
Mac OS 7 (Platform) 292
Mac OS 9 (Platform) 292
modifying methods 71
moving between pages 352–353
multi-page 350–354
naming 289
Never 128
object locking 289
object locking while modifying 268
opening 288–289
output 228–229, 480, 482
owner 265
placing fields on 367–368
previewing in Explorer 70
PRINT LABEL 512
printing pages of 362
printing subforms 509–510
role of 225
saving 362
using subforms 468–471
viewing pages in 362
Windows 3.11, NT 3.51 (Platform) 292
Windows 95/98/2000, NT 4 (Platform) 292
Forms page (Explorer) 69–71, 264, 288
Frames Assistant (Picture library) 666
functions 528
naming convention 533
role of 528

G

General Font 121
Generate (Typing methods) 699
Generate error file (Preferences) 138
Generate the Symbol File (Preferences) 138
Generated software packages (MacOS X) 719
Generic Web User (Preferences) 143
Get database parameter 116
Go to Line Number (Method editor) 585
GOTO PAGE command 354
used with Tab controls 448
graphic objects
changing appearance of 334–345
role of 230–231

specifying border patterns 343
specifying fill patterns. 342
specifying line width. 341
graphics, on labels. 493
greater than (>) character (in embedded fields and variables) 494
Grid Definition dialog box 320
grid, alignment. *See* alignment grid
grids
creating 459–460
displaying 271
Group Box tool/icon 275
Group menu item 280
Object menu 316
Group Owner (assigning) 620, 627
grouping fields (Form Wizard) 241
grouping objects 315–316
groups
access to 4D Client Web server. 636
access to plug-ins 636–637
assigning access privileges to 614
assigning to database objects 631–637
assigning to Design environment. 621
assigning to forms. 633
assigning to menu items. 635
assigning to methods 634
assigning to record operations 631–633
assigning to table definitions 631–633
assigning to User environment. 622
assigning users to 627–628
creating 626
in access hierarchy 617–619
modifying 627
nesting. 617, 629–630
ownership of forms. 633
removing users from 628
renaming in Form Wizard. 242
reordering fields 242
saving and loading 630–631
Groups list
adding users 627–628
displaying user names. 624
removing users 628

H

Header areas
adding custom graphics to 489
additional 480
additional levels of headers. 507
explanation of 480, 507
in output form. 477, 479
in reports 507
Header control lines 492
deleting 501
in reports 500
help
adding to fields 174, 188, 393–396
adding to objects. 393–396
deleting a message. 395
entering text of 395
selecting a message 396
help message
deleting 395
in Field Properties window 188
renaming 395
selecting. 396
Help page (Object Properties window) 394
hexadecimal numbers
display formats for. 402
entering in the Method editor 560
Hide icon 687
Hide Line Numbers 556
Hide toolbar. 112
hierarchical lists. 438, 641
creating 648–649
expanding and collapsing. 64
hierarchical pop-up menus 438
variable type 139
Highlight buttons 422
Variable type 139
HTML
Documentation 77
Encoding 147

I

icon
for expanding/collapsing preview area. 65
Locking (in 4D Server). 86
icons
adding to menu commands 604
associating with a list item 641, 650–651
for different types of database. 30
for protected objects 84
showing in debugger. 93
using for field types (Preferences) 125
Import method (File menu). 55, 585

Inactive Web Process Timeout 146
Include 4D Passwords (Preferences) 143
Indentation (Preferences) 122
indentations (Method editor) 559
Indexed attribute 183–184
  adding 191
  Alpha fields 177
  assigned to primary key field 206
  in Field Properties window 372
  in specifying table relations 195
  related fields 200
indexes (automatic updating of) 184
Indexing
  Fast Mode 193
  Traditional Mode 193
Indexing process 674
indicators
  assigning values to 455
  dials 453
  establishing settings of 453
  getting values from 455
  rulers 453
  thermometers 453
  using methods 455
Inherited forms (definition of) 357
Initialize Local Variables (Preferences) 139
initiators
  in entry filter codes 381
Input Form check box (Explorer) 263
INPUT FORM command 263, 289
input forms 228
  definition 225
  lists in 643
  multi-page 350–353
  role of 227
  setting the current 262–263
Insert an OLE Object menu item 281
Inserting and Deleting Frames (Picture library) . . 668
instances
  of menus 602
  See also connected menus
  using to manage menus 590
Integer fields 178
interface
  drop area 112
  objects 365
  Preferences 109
intermediate tables (for Many to Many relations) 215, 216–220
interpreted mode 691
  difference between interpreted and compiled . 690
  speed of execution 691
Interpreted mode and compiled mode (Navigation) 708
interprocess arrays, in compiler methods (Preferences) 140
interprocess variables 702
  compiler methods (Preferences) 140
Invisible attribute 185
Invisible buttons 423
  Variable type 139
Invisible check box (Table Properties) 169
invisible fields 185
invisible methods 544
invisible tables 164, 169
IP Address (Preferences) 142
Items menu
  Delete menu item 650
  New menu item 647
  Sort menu item 652

K

key combinations, reserved 608
Keyboard layout 392
Keyboard navigation (Method editor) 576
keyboard shortcuts
  assigning to active objects 418
  assigning to menu commands 597
  for clipboards 571
  in Property List 287
  navigational 576
Keys button (on Variable page of Object Properties window) 418
Keywords 554
  Method editor 550
  Preferences 124

L

Label editor
  using 515
  using label forms 516
label reports
  creating 513–516
  creating custom labels with graphics 493

printing 512–516
setting label width 513
setting margins 513
label width marker 269, 513
Labels menu item 516
language elements
assignment operator 532
commands 528
fields 528
flow of control structures 529
functions 528
object names 528
pointers 528
statements 529
variables 528
Large Size (Preferences) 121
Last Page button 424
Last Record button 424
layering
Down One Level 329
Move to Back menu item 329
Move to Front menu item 329
tools/icons 276
Up One Level 329
Layering tools/icons 330
leading zero 400
Leave Related Many Intact 206, 209
in Relation Properties window 205
less than (<) character
in embedded fields and variables 494
over maximum number of digits 400
Library pictures 346
licencelog.txt 719
License number (Integrating) 718
Licensed version 718
Limits (displaying) 272
Line check box (Menu Bar editor) 606
Line numbers (display) 556
line width (setting) 341
Line Width menu item 279
List editor 47–48, 642–654
creating Choice lists 188
creating lists in 646–650
Current List area 647
deleting items from 650
deleting lists from 650
Items menu 647
List of Lists 647
Lists menu 647
role of 48
sorting list items 652
List Editor menu item (Tools menu) 646
list form
created in Form Wizard 478
defined 225
for subrecords 469
viewing a record from 225
List of Menu Bars (Menu Bar editor) 594
list of open windows (in Design menu) 61
List of Tables window 515
LIST TO ARRAY command 436
lists
adding items 647
attaching to fields 188, 643
choice 374
creating 646–647
creating (Explorer) 82
default display (Method editor) 122
deleting 650
deleting items from 650
excluded values 645
making modifiable 653–654
maximum and minimum values 645
Method editor 550
modifying (Explorer) 82
non-sequential ranges 645–646
object locking 642
Ref ID 651
required 376
required values 645
role of 641–642
Show Lists 122
sorting items 652
specifying ranges in 652
used in methods 642
Lists menu
Delete menu item 650
New menu item 647
User Modifiable menu item 654
Lists page (Explorer) 82
Load Groups menu item (Passwords menu) 631
Local folder (Preferences) 129
local variables 702
in symbol table 703
initialization (compilation) 139
list 703
random values 139

location
macros . . . 578
temporary folder (Preferences) . . . 117
logical fields. *See* Boolean fields
Logweb.txt (Web Server) . . . 149
Long Integer fields . . . 178
look-ups. *See* wildcard (@) character

M

Mac . . . 292
Mac OS 7
Form Wizard . . . 249
Platform interface . . . 110, 292
Mac OS 9
Form Wizard . . . 249
Platform interface . . . 110, 292
Platform Interface (objects) . . . 335
Style sheet . . . 261
Mac OS X
pulsing buttons . . . 422
Mac OS X (style sheet) . . . 261
Mac Theme
Form Wizard . . . 249
Platform interface . . . 110
Tab controls . . . 449
machine language . . . 689
Macintosh
reserved key combinations . . . 607
scrolling . . . 62
Macros . . . 577
Calling . . . 581
Customized . . . 578
Default . . . 578
DTD . . . 761
Syntax . . . 579
Macros.xml . . . 578
mail-merge
creating documents . . . 493
using text areas for . . . 336
mail-merge documents. *See* form letters
Mandatory attribute . . . 182, 374
in Field Properties window . . . 372, 374
setting . . . 182, 373
Mandatory check box (Object Properties window) . . . 374
Mandatory Log File (Preferences) . . . 134
Mandatory value (Web server) . . . 148
manual relations . . . 197, 211
establishing . . . 208
multiple link relations . . . 222
reasons for using . . . 211
Many table . . . 197–199, 200
defined . . . 197–199
deleting records from . . . 205
displaying fields from . . . 256
removing relations . . . 209
Many to Many relations . . . 215
creating . . . 215–216
entering data in . . . 216–217
intermediate tables . . . 216–220
using intermediate tables . . . 215–216
Many to One properties . . . 201–202
Many to One relations . . . 197–199
Auto Assign Related Value . . . 204
creating . . . 207–209
entering data with . . . 211–212
Many table . . . 199
One table . . . 199
margins
setting in Form Properties . . . 297
setting in label reports . . . 513
marker labels (displaying) . . . 272
markers (displaying) . . . 272
marquee
for selecting objects . . . 305
for setting data entry order . . . 361
master table
defined . . . 227
selecting fields from . . . 237
master-detail relationship . . . 228
maximum and minimum values . . . 388
with lists . . . 645
Maximum Concurrent Web Processes (Preferences) . . . 146
Maximum Size (Data Segment Management dialog box) . . . 752
Maximum value (Web server) . . . 148
measurement scale, setting for form rulers . . . 312
Memory
Application Main Memory (Preferences) . . . 117
Block Size . . . 117
Cache (Preferences) . . . 116
Optimization . . . 116
Use New Memory Allocation Scheme on Macintosh . . . 116
Used Blocks . . . 117

Menu Bar editor. 45–46, 590, 593
adding menu items 596–597
adding menus 595–596
adding separator lines 606–607
assigning access privileges using. 635
assigning keyboard shortcuts 607–609
changing menu item font style 604–605
creating connected menus 602
creating menus 590
deleting connected menus 603
enabling and disabling menu items 605–606
enhancing menus 603–605
Menus list 677
modifying connected menus 602–603
Start a New Process check box 600
starting processes from 600, 676–678
Menu Bar Editor menu item 593
Tools menu 635
menu bars
creating 45, 593
default 592
in custom applications 611
object locking 591
previewing. 610–611
See also menus
menu commands
See also menu items
menu items
access privileges to 635–636
adding 596–597
adding separator lines to. 606–607
assigning groups to 635–636
assigning methods to 598–600
deleting 610
enabling and disabling 605–606
font style for 604–605
rearranging 598
menus
adding 595–596
adding menu items 596–597
connected 595
creating 591–597
creating connected 602
deleting 609–610
deleting connected 603
deleting menu items 610
designing 589–591
example of custom 589
for custom applications 611
instances of 602
length of title. 596
modifying connected 602
names as STR# resources 596
object locking 591
previewing 610
rearranging 598
reusing 601
Messages (Preferences) 121
Method editor 44–45
Allow Expand/Collapse 123
Default display. 122
Default Font. 122
Display conventions 560
Editing area 549
Keywords 550
Lists 550
managing long strings. 574
Options 122
Preferences. 121
shortcuts 576
Show Lists 122
Styles and color of syntax elements 124
syntax error checking in 564
Type-ahead 123
window 466, 548
method name
adding to a method. 75
Web Services 742
Method Properties dialog box 543, 635
Method Properties menu item (Method menu) . 634
Method templates 555
Method Type dialog box 466
methods
Allow Drag. 123
Allow Type-Ahead 123
arguments in 533
assigning access to. 634–635
assigning to menu items 591, 598–600
assignment operator (:=) in 529, 532
attributes 545
batch setting for method attributes 545
comments 571
compiler methods (Preferences) 140
creating 72
default font 122
definition of. 44
deleting 540
documentation 734

elements of 527–529
example 530–533
execution of 530–533
expand/collapse 123
exporting 55
find and replace 582
for indicators 455
Go to Line Number 585
importing 55
importing and exporting 585
indentation 122
inserting a command with its syntax 562
invisible 544
list 703
lists in the editor 550
modifying 73
name of published 735
naming 535
object locking 534
offered as a Web Service 545
operators in 528
parameters 735
properties (defining) 541
published as Web Services 729
published in WSDL 545
renaming 74, 542
renaming with 4D Server 543
role of 517
saving as a template 552
selecting components 562
showing line numbers 123
statements in 529
styles and color of syntax elements 124
syntax errors 564
types of 518–522
typing hexadecimal in 560
typing text in 560
using macros in 577
using process variables 531
using the wildcard (@) 567
using variables in 531
writing 559
Methods page (Explorer) 72–75
minimum height (hierarchical list) 656
Minimum value (Web server) 148
modifying
connected menus 602
field properties 190
form methods 71
lists 82
subform source forms 476
Move to Back menu item 279, 329
Move to Back tool/icon 330, 360
Move to Front menu item 279, 329
Move to Front tool/icon 331
Multi-homing (Web Server) 142
multi-page forms 350–354
adding page navigation tools 354
background page 345
creating with Form Wizard 351
printing 230, 350
Multiple copy-paste 570
multiple processes 228, 671–673
multiple table structures 156–158
Multiple Undo/Redo 569

N

Name of published method 735
Namespace 151, 733, 735, 743
naming
databases 28
fields 191
forms 65, 264
forms (Form Properties window) 289
groups (for forms) 242
methods 65
project methods 74, 542
tables 166
variables 531
negative numbers
displaying 400
formatting 401, 403
nesting (password groups) 629
Never (Forms) 128
New Child menu item (Items menu) 648
New Field menu item (Structure menu) 161
New Form menu item 235, 484
Design menu 69
New Method menu item 535
New Process check box (Execute Method dialog box) 679
New process function 600, 675, 682
New Table menu item (Structure menu) 164
New User menu item (Passwords menu) 624
Next error (Method menu) 697
Next Page button 424
NEXT PAGE command 528

Next Page tool/icon 353
Tools palette 351
Next Record button 424
No Action button 424
No Brace Matching menu item 562
non-enterable objects 420–421
number (#) sign
as a placeholder 400
in entry filter codes 377, 383
Number field formats 399–404
displaying numbers as times 402
displaying symbols and characters 400
examples 403
placeholders 400
positive, negative, and zero formats 401–402, 403–404
scientific notation 402
using custom formats 403
using decimal points 400
Number of HTTP requests 92
Numbering of clipboards 570
Numerics, default type (Compilation) 139

O

object events 525–527
Object List 304
choosing another object with 340
object locking
definition of 105
of table definitions 170
while modifying forms 268, 289
while setting preferences 107
with database objects 105
with field properties 176
with menus 591
Object menu 279–281
Align menu item 280
Align to Grid menu item 280, 322
Border menu item 279
Clear Object Method menu item 281, 467
Color menu item 279
Down One Level 329
Duplicate menu item 280, 326
Duplicate on Matrix menu item 280
Fill menu item 279
Group menu item 280, 316
Insert an OLE Object menu item 281
Line Width menu item 279
Move to Back menu item 279, 329, 331
Move to Front menu item 279, 329, 331
Object Method menu item 281
Show Format menu item 280
Show Resource menu item 280
Ungroup menu item 280, 316
Up One Level 329
Object Method button (Object Properties window) 465
Object Method menu item 281
object methods 462–519
attaching to fields or objects 533
clearing 541
creating 534
data entry management 519
deleting 467, 541
examples with fields 462
for controlling data 420
in output forms 482
in report with breaks 508
modifying 467
placing 533–534
role of 518, 518–519
uses of 462
viewing 304, 467
with fields and objects 462–467
writing 525–527
object names (referring to in methods) 528
object properties
displaying 271
overview of 231–232
viewing 303
Object Properties window . . 260, 340, 343, 366, 412
Colors page 341, 343, 344
Coordinates page 306, 310
creating object methods with 518
Data Control page 388, 389, 397
Display page 336, 427
Enterable attribute 373
Events page 437, 463
Font page 340
Help page 394
lists, *See* choice lists
Mandatory attribute 373–374
Mandatory check box 374
print frame options 509–510
Print Variable Frame option 511, 512
setting data entry controls 417
setting display formats 415

setting entry filters 376–385
setting font attributes with 339
specifying a custom format 405
Subform page 470, 475
used to set appearance option 259
Variable page 418, 440
objects
3D radio buttons 427–429
active 230–231, 419–455
aligning 316
border patterns 343
button grid 439–440
buttons 421–425, 427–429
changing the appearance of 334–345
check boxes 426–427
combo box 438
copying on forms 328–329
creating active 414–419
creating in a series 459
creating with Tools palette 314
deleting 331
deselecting 303
dials 453–455
display formats for 415
drawing 312–314
duplicating 326
enterable 420–421
fill patterns for 342
graphic 230–231
grouping 315–316
hierarchical list 438
hierarchical pop-up menu 438
layering 329–331
moving 305
non-enterable 420–421
nudging 305
picture menu 441–444
plug-ins 450–452
pop-up menus 435–437
radio buttons 427–429
radio pictures 427–429
resizing 307–309
resizing handles 302
rulers 453–455
scrollable areas 435–437
selecting 302–305
selecting multiple 304–305
setting color of 344
setting line widths of 341
tab control 445–448
thermometers 453–455
ungrouping 316
using object methods with 462–467
Offered as a Web Service 545, 730
Old Edit menu mechanism 592, 595
OLE_Tools
Commands 555
Plug-in areas 451
On Activate event 525
On After Keystroke event 463, 526
On Before Keystroke event 463, 526
On Clicked event 439, 463, 525
On Close Box event 526
On Close Detail event 526
On Data Change event 464, 526
On Deactivate event 525
On Display Detail event 464, 526
On Double Clicked event 463
On Double-clicked event 525
On Drag Over event 464, 526
On Drop event 464, 526
ON ERR CALL method 617
On Event Manager process, ... 674
On Exit 524
On Exit database method 709
On Getting Focus event 464, 526
On Header event 464, 526
On Load event 447, 463, 525
On Losing Focus event 464, 526
On Menu Selected event 526
On Open Detail event 526
On Outside Call event 464, 525
On Plug-in Area event 464, 526
On Printing Break event 464, 527
On Printing Detail event 464, 527
On Printing Footer event 464, 527
On Resize event 526
On Server Close Connection 524
On Server Open Connection 524
On Server Shutdown 524
On Server Startup 524
On Startup 524
On Startup database method 709
On Timer event 464, 526
On Unload event 463, 525
clearing an array with 448
On Validate event 463, 525
On Web Authentication 524

On Web Connection 524
One table 197–199, 200, 201
  defined 197–199
  deleting records in 205
  entering data in 211–212
One to Many properties 203–204
One to Many relations
  See also Many to One relations
One to Many relations, using subforms in 468
One to One relations 214
On-line documentation (Access from the Explorer) 77
On-line Help 301, 759
  Calling from a 4D Database 759
Open Compiled 32, 708
Open Data File dialog box 34–35
Open Interpreted 32, 708
Open menu item 24, 29
Open Subform button 425, 471
Open window function 297
opening
  databases 29
  form methods 538
  forms (Form editor) 288
  methods 534
  options 31
  project methods 535
Opening mode 32
operators (in methods) 528
Option-clicking (to open object method) 465
Options
  Compiler 698
  Method editor 122
  Preferences 127
  Preferences (Web) 148
Options page (Form Wizard) 245
ORDER BY command 528
output control lines 269, 329, 478
  definition of 480–482
  enabling 482
  moving 483
  removing 484
output control markers 478, 483
  in Form editor 483
Output Form check box (Explorer) 263
OUTPUT FORM command 263, 289
output forms 228–229
  control lines 480–482
  creating mail-merge documents 493–495
  definition 225
  designing labels with 493
  for reports 491
  labels 515
  modifying 489
  setting the current 262–263
  uses of 477
Owner
  setting for forms 634
  setting for groups 620
  setting for methods 634, 635
  setting for objects created by user 626
Owner drop-down list
  Form Properties window 634
  function of 543
  in Form Properties window 291
  in Method Properties window 635
owner privileges
  for forms 291, 633
  for groups 620
  for methods 634

P

Page 0 objects (displaying) 271
page border lines 362
Page buttons 424
page navigation actions 424
page navigation tools 354
Page pop-up menu 353
Page Setup dialog box 362, 497
page tools/icons 276
Panes (Method editor) 557
Paper (displaying borders in Form editor) 271
parameters
  advanced (Web Services Wizard) 742
  for Picture button 431
  for picture pop-up menu 443
  for templates 555
  table (for published method) 743
  using advanced (proxy methods) 741
parent records (in subtables) 181
parent tables 158
parentheses (()) (as dead characters) 384
Password Access editor 46, 624
  assigning users and groups 624–626
  changing access groups with 627
  changing user name or password 626
  Groups list 624

role of 46, 613
Users list 624
viewing database usage 638
password access system 614–639
adding users 624–626
Administrator 619
assigning groups to forms 633
assigning groups to menu items 635–636
assigning groups to methods 634–635
assigning groups to plug-ins 636–637
assigning groups to record operations 631
assigning groups to table definitions 632–633
assigning passwords to users 625
assigning users to groups 627–628
creating access groups 626
creating an access hierarchy 617–619
Designer 619
disabling 623
error messages 617
group owners 620
initiating 623
maintenance of 637
nesting groups 629
object locking in 4D Server 614
removing nested groups 629
removing users from groups 628
role of 614
saving and loading groups 630–631
user hierarchies 629
viewing usage of 638–639
passwords
access to the User mode 621
case sensitivity 625
changing 626
creating 625
Default User 622
displaying the password entry dialog box 134
displaying users 134
entering 615–616
include 4D Passwords (Web Server) 143
sorting list of users by alphabetical order 134
using (Web Server) 142
Passwords menu 624
Edit Group menu item 627
Edit User menu item 626
Load Groups menu item 631
New Group menu item 626
New User menu item 624
Plug-ins Access menu item 636
Save Groups menu item 631
Passwords menu item (Tools menu) 624, 636
Paste (standard action) 425
Paste menu item 57, 329
paths (to data files) 35
Pause icon 685
Physical files 28
Picture button (parameters) 431
picture buttons 430–435
variable type 139
Picture check box (List editor) 651
Picture fields 179
formats for 409–411
modules in 180
on background 410–411
plug-in areas in 180
printing 511–512
scaling to fit 410
storing data from plug-ins in 180
truncating 409
Picture library 48–49, 345, 430, 443, 659–670
adding Pictures 661
creating custom toolbar with 604
creating thumbnails 666
for storing background graphics 411
graphic tools 665
placing a picture on a form 345–347
shortcuts 665
storing small icons in 650
using to add pictures to forms 234
picture menu 441–444
Picture pop-up menu (parameters) 443
Picture pop-up menus (Variable type) 139
Picture radio buttons (Variable type) 139
pictures
adding 661
editing 664
Picture library (modification) 659
setting properties 663
viewing 660
placeholders
asterisk (*) 400
caret (^) 400
in Alpha field formats 405
in entry filters 381, 383

in number field formats 400
number (#) sign 400
underline (_) character 383
zero 400
Plain text 124, 558
platform
compilation 137
preferences 110
Platform Interface
Automatic 249
Automatic option 99, 292, 335
Best Match 110
for objects 98, 335
Inherit from Database option 292
Inherited from Form option 335
Mac OS 7 249
Mac OS 9 249
Mac Theme 249
Macintosh option 335
objects affected by 99–104
setting for forms 292–293
setting for text objects 339
specifying font attributes 259
specifying in Form Wizard 244, 258
Windows 3.11, NT 3.51 249
Windows 95 option 335
Windows 95/98/2000, NT 4 249
Windows NT 3.5.1 option 335
Platform Interface drop-down list (Form Properties window) 293
Platform Interface drop-down list (Preferences) . . .98
Plug-in areas
4D Chart 451
OLE tools 451
Report 451
plug-in objects 450–452
plug-ins
4D View 452
assigning access to 636–637
assigning groups access to 636
using 452
pointers
arrow 302
crosshair/crossbar 313
entry order 359
multi-directional arrow 307
pop-up menus
definition (as active objects) 435
hierarchical 438
initializing 390, 435
picture 441
shortcuts for inserting 670
Pop-up/drop-down lists (Variable type) 139
PostScript™ printer, printing graphics 179
power failures 36
Preferences 57, 124
4D Server 118
Access Management 132
Activate Automatic Comments 131
All variables are typed 140
Allow 4D Open Connections 120
Allow 4D WebSTAR to connect via 4D Connect 149
Allow Deletion Control 135, 209
Allow Drag 123
Allow Expand/Collapse 123
Allow Read Only Data file Use 136
Allow SSL for Web Server 142
Allow Type-Ahead 123
Application 114
Application Main Memory 117
Automatic Comments 88
Automatic Form Creation 128
Automatic Transactions during Data Entry . . 135
Cache 145
Client/Server 118
Client/Server Connections Timeout 119
Color Background 125
Color Names 125
Comments 131
Compatibility 128
Compilation 137, 700
Compilation Path 139
Compiler Methods 140
Configuration (Web) 145
CPU Priority 115
Data Control 134
Database 132
Database Cache Memory 116
Default Alpha Type 139
Default Button Type 139
Default Font 121
Default Home Page 145
Default HTML Path 144
Default HTML Root 144
Default User 622
Design Mode 120
dialog box (default environment) 31

Display User List in Password Dialog Box 134
Displaying toolbar 112
Documentation 129
drop area 112
Editing filters (Web Server) 147
Encrypt Client/Server Connections 119
Exit Design Environment when going to Custom Mode 127
FichiMandatory Log File 134
Flush Data Buffers Every ... Minutes 116
Fonts 120
Formats and filters 113
Formatting 113
General Font 121
Generate error file 138
Generate the Symbol File 138
Generic Web User 143
Indentation of methods 122
Initialize Local Variables 139
Interface 109
IP Address 142
Large Size 121
Maximum Concurrent Web Processes 146
Message Font 121
Method Editor 121
Method editor font 122
Options 127
Platform 110
Platform Interface option 98
Print Titles 126
Process and interprocess are typed 140
Progress Indicator 112
Proxy address 152
Proxy Port 152
Publish Database at Startup 142
Publishing 141
Range Checking 138
Register Clients at Startup For Execute On Client 119
Regular Size 121
Reuse Temporary Contexts 144
Save Request in File 149
Setting (Compilation) 137
Show Line Numbers 123
Standard Set 147
Starting Mode 143
Startup Environment 127
Structure Access 132
Structure Editor 125
Style Sheets 114
System Resources 115
TCP port 141
Text Conversion 147
Toolbar 112
Type Default Numeric Type 139
Type the variables 139
Use Icons for Field Types 125
Use New Memory Allocation Scheme on Macintosh 116
Use V3.x.x File Procedure Scheme 129
Use V3.x.x Startup Method Scheme 128
User List in Alphabetical Order 134
User Mode Access 133
Web 141
Web Passwords 142
Web Process 146
Web Services 150, 744
Web Services Namespace 151
Wizard Method Prefix 152
Preview area icon (Explorer) 68
Preview on screen option (for printing reports) . 497
Previous error (Method menu) 697
Previous Page button 424
Previous Page tool/icon 353
Previous Record button 424
Previous/Next (Method editor) 583
primary key field 200, 201, 207
definition of 195
required attributes for referential integrity . . . 206
PRINT LABEL 512
Print Variable Frame check box
printing subform areas 509
printing text fields 512
print window titles 126
printing
form pages 362
labels 512–516
Picture fields 511–512
Print Titles (Preferences) 126
subform areas 509–510
Text fields 512
Privileges page (Table Properties) 633
process
$4D Compiler 674
$Runtime explorer 674
elements of a 672
Process and interprocess are typed (Preferences) . 140

Process and interprocess variables
List 702
Typing 140
Process list 679–688
Process List editor 679–688
viewing processes 679–684
window 680
Process menu 685
bringing a process to the front 688
pausing a process 685
resuming execution of a process 685
tracing a process 686
Process Page (Runtime Explorer) 93
process variables 702
processes
aborting 681, 685–686
bringing to front 687–688
controlling execution of 685
debugging 686
default 673
default names of 682
definition of 671
execution of multiple 674
for managing a multi-window interface 671
hiding 686
multiple 671–673
naming 681–682
pausing and resuming 685
process number 681
process status 683
process time 684
role of 671
specifying when to execute 675
Start a New Process check box, in Menu Bar editor 677
starting from a menu item 676–678, 682
starting new 678
starting using Execute Method 678–679, 682
starting with a method 675–676, 682
tracing 686
Progress Indicator (Preferences) 112
project methods
assigning to menu items 591, 598
batch setting of attributes 75
creating 72, 535
in symbol table 703
renaming 542
role of 518, 522
Prompt if Related One does not exist check box (Relation Properties window) 202, 212
properties
inherited forms 355
of subforms 475
Property List
Displaying 271
Object Types 370
Proxy Address (Preferences) 152
Proxy methods 738
Calling 745
Proxy Port (Preferences) 152
proxy_ 152
Publish Database at Startup (Preferences) 142
Published in WSDL 545, 731
Pulsing buttons 102

Q

Quartz (Preferences) 111
Queries (new Index mode) 192
Query editor 372, 374
quick reports (creating with Many to Many relations) 220
QuickTime™ compression 186
Quit
Standard action 426
Under MacOS X 426, 601
quotation marks ("), in entry filters 382

R

radio buttons 427–429
formatting Boolean fields as 179
naming 428
Variable type 139
radio pictures. *See* radio buttons
range checking
diagnosing anomalies 707
disabling locally 706
Range Checking (Preferences) 138
range of characters, in filter arguments . . . 382–383
ranges, in lists 645–646
Real fields 178
Record Access area (Table Properties) 167, 633
record navigation actions 424
record operations 632
records 154
Flush Data Buffers Every ... Minutes 116

recovering by tags (using the Completely Delete option) . . . 170
Redo (Edit Menu) . . . 57
Ref ID (list element) . . . 651
referential integrity . . . 205, 372
  Can't Modify attribute . . . 206
  Indexed attribute . . . 206
  Unique attribute . . . 206
Register Clients at Startup For Execute On Client (Preferences) . . . 119
Regular Size (Preferences) . . . 121
Related Fields area (Relation Properties window) 201
related One table, selecting fields from . . . 238
related records, deleting . . . 205
related tables . . . 193–195
  Can't Delete if Related Many option . . . 205
  creating table relations . . . 200–209
  Delete Related Many option . . . 205
  deleting records in . . . 205–206
  entering data in . . . 211–212
  example of . . . 194
  Leave Related Many Intact option . . . 205
  One table and Many table . . . 197–199
  related fields . . . 195–197
  relation types . . . 214–220
  unusual relation types . . . 214
  wildcard look-ups in . . . 212–214
relating tables . . . 200–209
relation line
  drawing the . . . 201
  setting color of . . . 209
relation properties
  modifying . . . 210
  setting . . . 200–206
Relation Properties window . . . 208
  Color page . . . 209
  Control page . . . 208
  for enforcing referential integrity . . . 372
  modifying relations using . . . 210
  overview of . . . 201
  Related Fields area . . . 201
relation types
  Many to Many . . . 215
  Many to One . . . 197–199
  One to One . . . 214
relational databases (definition) . . . 157
relations
  Allowing Deletion Control . . . 135
  analyzing . . . 220–224
  between tables . . . 38, 194
  circular . . . 221–222
  creating . . . 200–209
  displaying . . . 157
  foreign key field . . . 195
  multiple links . . . 222–223
  multiple record . . . 223–224
  primary key field . . . 195
  re-establishing . . . 210
  removing . . . 209
  See also related tables
Remove this list (Method editor) . . . 551
renaming forms . . . 264–265
  in Explorer . . . 65
renaming methods . . . 74
  in Explorer . . . 65
Replace
  Edit Menu . . . 57
  in methods . . . 584
Replace Next (Edit Menu) . . . 57
Replace Previous (Edit Menu) . . . 57
Report (Plug-in areas) . . . 451
reports
  break levels in . . . 497–499
  columnar . . . 491
  creating . . . 496–497
  creating custom labels . . . 493
  creating from Many to Many structures . . . 220
  creating label . . . 513–516
  initiating break processing . . . 501
  object methods . . . 508
  one record per page . . . 492
  subforms in . . . 492
  subtotals in . . . 501–503
  summary . . . 505
  Text fields in . . . 492
  with breaks (example of) . . . 506
repositioning options
  automatic . . . 309
  in Object Properties window . . . 311
Request log . . . 149
required lists . . . 376
reserved key combinations . . . 607
reserved words (in field names) . . . 174
Resizable check box (Form Properties window) . . 299
resizing
  automatic . . . 309
  forms . . . 297
  grouped objects . . . 315

handles 302, 307
objects, with the arrow keys 308
objects, with the resizing handles 307
options (Form Properties window) 299
table images. 161
Resume icon 685
Reuse Temporary Contexts (Preferences) 144
Revert to Saved menu item (File menu) 363
rgrg (random value) 139
RPC 727
Ruler Definition dialog box 312
rulers 311–312, 453–455
assigning values to 455
changing units of 269, 311
displaying 271
example of. 453
getting values from 455
setting measurement scale 312
with output control markers. 483
Rulers menu item 311
Run compiled 58
Run interpreted 58
Runtime Explorer 90
$4D Compiler 674
Break page 93
Catch page. 93
Process page. 93
Watch page 91

S

Save As Template. 555
Save *EditorName* menu item 105
Save *FormName* menu item 362
Save Groups menu item (Passwords menu) 631
Save Request in File (Preferences) 149
saving
forms 362
groups 630
modifications to pictures 666
password access groups 630–631
Scale command (Form menu) 333
Scaled to Fit picture formats 410
scientific notation, in number field formats 402
Scrapbook (storing copied objects) 328
Screen Sizes (Form Size). 253
scroll bars. 62
adding to Text fields 178, 392
scrollable areas (initializing) 435
scrolling 62
in forms 362
table images 161
Searching Options (Find Editor) 97
Security of Web Services 726
security. *See* password access system
segmenting
data files. 749, 750–756
existing data files. 750
Select a Menu dialog box (Menu Bar editor). . . . 602
Select All (standard action) 425
Select All menu item 304, 329
in Edit menu 57
Select Enclosing Block (Method editor) 572
Selectable check box (for subforms) 470, 474
Selected Process (Runtime Explorer) 91
selecting
4D Engine folder 718
a different data file 32
fields for forms. 237–240
method components 562
objects 302–305
selection rectangle. *See* marquee
semicolon (;) character
in number field display formats 401
specifying display formats for embedded fields and variables 495
Send Extended Characters Directly (Preferences) 147
separator lines, adding to menu items 606–607
sequence numbers
as default values. 390
generating as default values 390
in related tables 196
Sequence of Frames (Picture library) 667
series (creating objects in) 459
SET DATABASE PARAMETER. 116
SET PLATFORM INTERFACE command 99
Setting CPU Priority. 115
Setting Page (Preferences) 137
Shift key (to constrain object shape). 313
Shift+Alt clicking (for choosing tables and fields) 495
Shift-clicking (for selecting objects) 304
Shift-dragging (moving output control lines). . . 483
shortcuts
inserting fields in text areas 495
inserting Picture Buttons. 670
inserting Pop-up Menus 670

keyboard, for buttons and check boxes 418
Method Editor 576
Picture Library 665
Shortcuts dialog box 418
Show Clipboard (standard action) 426
Show Clipboard menu item 57
Show Custom Menus menu item (Menu menu) . 610
Show Field and Table Numbers (Runtime Explorer) 93
Show Format menu item 280
Show Icons (Runtime Explorer) 93
Show Integers in Hexadecimal (Runtime Explorer) 93
Show Line Numbers 556
Preferences 123
Show Lists (Method editor) 122
Show lists (Preferences) 552
Show Name menu item (Objects menu) 280
Show Resource menu item 280
Show Toolbar (Preferences) 112
Show Types (Runtime Explorer) 93
Show warnings (Compiler) 697
Showing/Hiding Elements in the Form Editor . . 270
single-table structures 156
Size Based On drop-down list (Form Properties window) 298
sizing options
for forms 297
Object Properties window 311
SOAP (Definition) 724
Soap Action (Web Service parameter) 742
SOAP requests (values needed) 735
SOAP Type 743
Sorted Tables and Fields (Runtime Explorer) 93
sorting
list of users (Preferences) 134
lists (List editor) 652
records (for break processing) 499
spaces
as dead characters 384
in number display formats 401
Special menu (for executing methods) 678
splash screens
adding 610
default size 611
Split button (Data Segment Management dialog box) 756
splitters 456
Automatic 458
examples 458
interaction with neighboring objects 457
managing programmatically 457
Splitting the editor window 557
SSL (Encrypting Client/Server Connections) 119
standard actions 424
Standard Set (Preferences) 147
Start a New Process check box (Menu Bar editor) 600, 677
starting 4th Dimension 24
Starting Mode (Preferences) 143
Startup Environment (Preferences) 127
startup method
assigning to a user 626
Use V3.x.x Startup Method Scheme option . . . 128
STARTUP procedure 128
statements (in methods) 529
STR# reference 596
strings
managing (Method editor) 574
types 96
Structure Access (Preferences) 132
Structure Access drop-down list (Preferences) . . . 621
Structure editor 42, 153
creating fields 173–175
creating tables 164–171
definition of 160
fonts 121
Preferences 125
renaming tables 167
structure files 32
Structure menu
creating tables 164
displaying table properties 165
Structure window 42
Contextual menus 60, 282
creating table relations 207–209
for new databases 160
reestablishing relations 210
removing relations 209
showing relations 209
table images in 160
Style menu 281
Style Sheet Definition dialog box 261–262
Style Sheet drop-down list (Form Wizard) 262
Style Sheet editor 259–262
accessing 260
style sheets
applying 262
creating 260–262

in Form Wizard 244
Preferences. 114
setting font attributes with 340
Styles (methods) 558
Styles and color of syntax elements 124
Styles page
Appearance drop-down menu 258
in Form Wizard 244, 258, 260
subfields. 180
Subform
Opening source form 476
Subform page
in Form Wizard 237, 246
Object Properties window. 470, 475
Subform tool/icon 275
subforms 217–220, 468–476
adding to a form 71, 255
Auto Assign Related Value 203
Automatic Width check box. 475
buttons for. 425, 471
creating 471–475
creating records in. 220
creating using Form Wizard 234
defined 228
displaying information in. 217–220, 256, 471
entering data in. 220, 469
entering data into a. 211
modifying source forms 476
printing 509–510
printing with fixed-frame option 510
printing with variable option 510
properties 475
specifying in Object Properties window 470
using 468–471
using default input forms 471
using multiple 468
with reports 492
subrecords 158, 180
Subscribing to a Web Service 736
subtables 158–159, 180–181, 237
adding subform from 468
adding with Form Wizard. 246
parent records in 158, 181
subfields in 180
subrecords in 180
when not to use 159
Subtotal function (initiating break processing with) 500
subtotals (computing in a report) 508
summary calculations
in reports 477
printing 482
summary reports 505
Swap expression. 573
Symbol File 138
symbol table (list of local variables) 703
syntax elements
customizing 558
styles 124
syntax errors (checking and correcting) 564
System Resources (Preferences) 115

T

tab control 445–448
as page navigation tool 354
with icons and text 446
Tabable property 417
table image
moving. 163
previewing in Explorer 68
resizing. 161, 162
scrolling 161
setting the color of 171
viewing in Explorer 68
table methods (role of) 518, 519
table names
duplicate 167
Dynamic 370
maximum number of characters. 167
updating in methods. 166
Table Properties menu item (Structure menu) 161, 165
Table Properties window 165, 166–171
Completely Delete property 170
invisible table 169
Owner 167
Privileges page 633
Record Access area 167, 633
table relations
advantages of. 194
analyzing 221
circular. 221
Many to One 197
See also related tables
tables 155–159
assigning access to. 632
canceling creation of 164

creating . . . 164–165
definition of . . . 155
master . . . 227
maximum number of . . . 158
moving images . . . 163
naming . . . 167
parent . . . 158
relating . . . 193–209
renaming . . . 166
selected . . . 161
selecting images of . . . 161
setting name color . . . 171
setting properties . . . 165
subtables . . . 158–159
Tables page (Explorer) . . . 67, 161, 369
tags (recovering by) . . . 170
target width (specifying in Form Wizard) . . . 489
TCP port (Preferences) . . . 141
templates . . . 236
creating a form . . . 258
databases . . . 26
for methods . . . 552
in Form Wizard . . . 233
parameters . . . 555
Temporary Contexts (4D Client Preferences) . . . 144
Temporary files (Preferences) . . . 117
Temporary folder (Preferences) . . . 117
text (entering in methods) . . . 560
Text Area tool/icon . . . 275, 338
for form letters . . . 494
text areas
as graphic objects . . . 231
changing the appearance of . . . 336–340
creating on forms . . . 336–338
embedding fields and variables in . . . 336, 493–495
setting appearance . . . 339
setting default font attributes . . . 338
setting platform interface . . . 336
variable frame option . . . 509
text attributes (setting for text objects) . . . 339
Text Conversion (Preferences) . . . 147
Text fields . . . 177–178, 336
adding scroll bars to . . . 392
in reports . . . 492
printing with variable frame . . . 512
thermometers . . . 453–455
as progress indicators . . . 453
example of . . . 453
Preferences . . . 112
setting properties of . . . 455
tilde (~) character
forcing capitalization . . . 381
in entry filter codes . . . 377
initiating an entry filter . . . 381
Time fields . . . 179
display formats for . . . 398–399
time-sliced execution (of processes) . . . 674
toolbar
creating a custom . . . 604
Design environment . . . 62–63
displaying (Preferences) . . . 112
showing . . . 112
Tools menu . . . 59
Explorer item . . . 63
Menu Bar Editor menu item . . . 593
Tools palette . . . 273–276, 312
Add Field tool/icon . . . 275, 367
Alignment tools . . . 276
alignment tools/icons . . . 317–318
Arrow tool/icon . . . 275, 302
displaying . . . 271
Distribute tools . . . 322
drawing objects . . . 312
Duplication tool . . . 276
in Form editor . . . 273–276
Layering tools/icons . . . 276, 330
Next Page tool/icon . . . 351
Next Page/Previous Page tools/icons . . . 353
Object method deletion tool . . . 276
Previous Page tool/icon . . . 352
Subform tool/icon . . . 275
Text Area tool/icon . . . 275, 338
totals
calculating in reports . . . 491
object methods for calculating . . . 508
Trace icon . . . 686
Traditional Mode (new Index mode) . . . 193
trailing zero . . . 400
triggers
creating . . . 72, 537
data validation . . . 366
enforcing business rules . . . 366
for enforcing business rules . . . 366, 372
role of . . . 518
setting for a table . . . 167
writing . . . 524–525
Triggers page (Table Properties window) . . . 523

truncated field names 174
truncated picture format 409
Turn Grid On/Off menu items 278
Type drop-down menu (Object Properties window) 423
Type the variables (Preferences) 139
Type-ahead
  Disable 123
  enabling/disabling 123
  use in Method editor 566
  use with macros 581
Typing methods 699

U

UDDI (Definition) 724
Uncomment 572
underline (_), for placeholders 383
Undo (standard action) 425
Undo menu item 332
unexpected interruptions 36
Ungroup menu item 280, 316
  Object menu 316
Unique attribute 184–185
  assigned to primary key field 206
  changing field properties 191
  in Field Properties window 372
  in table relations 195, 196
Unlimited (Client/Server Timeout) 119
Unlimited (Web Process Preferences) 146
unrelated tables (adding fields from) 239
Up One Level 329
Update Time (Runtime Explorer) 91
Updates (new Index mode) 192
URL
  Access to a Web Service 739
  Access to a Web Service published by 4D 735
Use 4DVAR Comments instead of Brackets (Preferences) 148
Use Icons for Field Types (Preferences) 125
Use Javascript for Entry Control (Preferences) . . .148
Use menu 57
  Custom Menus item 40
  Design item 40
  for changing environments 38, 40
  User menu item 38
Use new context referencing mode (Preferences) .148
Use New Memory Allocation Scheme on Macintosh (Preferences) 116
Use Passwords (Preferences) 142
Use V3.x.x File Procedure Scheme (Preferences) . 129
Use V3.x.x Startup Method Scheme 521, 525
  Preferences 128
Used Blocks (Preferences) 117
User (standard action) 426
User Defined (Web character set) 147
User dialog box 625
User environment 39
  changing input and output forms in 263
  defining group access to 621
  List editor in 654
  role of 39
  testing report forms 497
User Mode Access (Preferences) 133
User Modifiable menu item (Lists Menu) 654
User/Custom Menus process 673
users
  adding 624–626
  assigning passwords to 624–626
  assigning to groups 627
  changing names 626
  Default user 133
  displaying in the Password dialog box 134
  removing from groups 628
  sorting the list 134

V

values
  maximum and minimum 388
  setting default 389–391
variable frames
  option for printing picture fields 511
  option for printing Text fields 512
  printing subforms 509–510
Variable page (Object Properties window) . 418, 440
variables
  associated with buttons 423
  compiler methods (Preferences) 140
  embedded in text areas 493
  for temporary data storage 420
  in mail-merge templates 493
  in methods 531
  naming of in methods 531
  process 531
  Runtime Explorer 91

vertical bar (|) character 385
Vertical Scroll Bar check box 393
volumes (storing segments on) 749

W

Warning messages 707
warnings 705
  showing 697
Watch Page (Runtime Explorer) 91
Web
  Connections (configuration) 146
  Preferences 141
  Publish Database at Startup 142
  Send Extended Characters Directly 147
  SSL 142
  Standard Set 147
  Text Conversion 147
Web browsers 228
Web cache occupation (Runtime Explorer) 92
Web configuration (Preferences) 145
Web file to send (Runtime Explorer) 92
Web Filters (Preferences) 147
Web Passwords (Preferences) 142
Web Process (Preferences) 146
Web server
  4D Client (license) 636
  cache 145
  Clear Cache 146
  compatibility (Preferences) 148
  Default Home Page 145
  Default HTML Root folder 144
  Generic Web User 143
  Include 4D Passwords 143
  IP Address 142
  managing access to 4D Client 637
  passwords 142
  request log 149
  starting mode 143
  Using Javascript 148
  using passwords 142
Web server activity time (Runtime Explorer) 92
Web Server Publishing (Preferences) 141
Web Service Name (Preferences) 151
Web Service name (SOAP requests) 735
Web Services 52, 723
  Access (Server) 734
  Allowing requests 151
  Creating methods for 729
  Documentation (Server) 734
  Integration in 4D 724
  Name (Server) 732
  Offered as a Web Service 545
  Parameters 743
  Preferences 150
  Published in WSDL 545
  Publishing 728
  Security 726
  Subscribing to 736
  Wizard 52, 737
  Wizard Method Prefix 152
Web Services Namespace (Preferences) 151
Web site (Preferences) 129
WebFolder (Default HTML Root) 144
WEDD resource 36
Whole Object Name (Find Editor option) 97
Whole Word 584
Whole Word (Find editor option) 583
wildcard (@) character
  in entry filters 379
  Method editor 567
  searches in related tables 212–214
  using 567
Wildcard Choice list
  in Relation Properties window 204, 208
  using in data entry 212–214
windows
  expanding or closing 61
  from several environments 41
  making active 61
  moving 61
  moving between 61
  printing titles (Preferences) 126
  setting margins 297
  setting titles 295
Windows 3.11, NT 3.51
  Platform Interface 110, 292
  Platform Interface (Form Wizard) 249
Windows 95/98/2000, NT 4
  Platform Interface 110, 292
  Platform Interface (Form Wizard) 249
Windows 98/2000 (Style sheet) 261
Windows XP (Style sheet) 261
windows, editor. *See* editor windows
Wizard Method Prefix (Preferences) 152

WSDL
Creation. 151
Definition 724
Discovering 739
Generation of 731
Published in WSDL option 545

X

XML (Namespace) 151, 733
x-user-defined (Web character set) 147

Z

zero
as a placeholder. 400
in number fields 401